



LUND UNIVERSITY
School of Economics and Management
Department of Informatics

Platform Thinking

Considerations for establishing third-party developer-centric ecosystems

Master thesis 15 HEC, course INFM10 in Information Systems

Authors: Björn Arvesen
Sylvia Karanja

Supervisor: Bo Andersson

Examiners: Odd Steen
Miranda Kajtazi

Platform Thinking: Considerations for establishing third-party developer-centric ecosystems

AUTHORS: Björn Arvesen and Sylvia Karanja

PUBLISHER: Department of Informatics, Lund School of Economics and Management,
Lund University

PRESENTED: June, 2018

DOCUMENT TYPE: Master Thesis

NUMBER OF PAGES: 120

KEY WORDS: platform business models; boundary resources; third-party developers; platform governance; developer marketing; external innovators; lean communication; complementors; integrators; organizational identity.

ABSTRACT:

There has been a dramatic growth in organizations aiming to leverage the power of platform business models and capture the vast amounts of value stemming from external innovation. Thus, actors are required to open up their innovation processes to third-party developers, and make sure these external resources are incentivized to both enter and remain in the ecosystem. So far, there has been a limited understanding of what considerations have to be made to create an optimal developer environment, with research being limited to specific aspects of platform establishment. By applying a process perspective, this study has identified the three high level phases of such platform establishment as (1) Understanding Developers, (2) Platform Design and (3) Governance and Control, which all come with certain considerations. The process has been examined through six interviews with different experts in the field, to create a comprehensive list of the key considerations for each phase. Using diffusion of innovation theory, the study does not only list various factors, but also explains their impact on platform diffusion. Key findings highlight discrepancies between literature and empiricism, with addressable market and brand equity of a firm being the primary motivations for third-party developers. Additionally, the study identifies strategic partnerships as an impactful factor in all phases and a necessity for successful platform establishment.

Acknowledgements

We wish to thank all participating organizations and informants for taking part in our study. Your insights have been invaluable for this study and we truly appreciate you taking the time to talk to us. Our sincere gratitude as well and many thanks to our supervisor Bo Andersson, who with excellent support, encouragement and guidance even when unwell made the creation and completion of this thesis possible. Throughout our research, Bo provided us with rewarding discussions and constructive criticisms that challenged us to produce our best. Thank You!

Special Thanks

First, I would like to thank God for good health and His mercies always. To the Swedish Institute Study Scholarship for the financial support of my studies. Sincere thanks to my family and especially my mom, dad and sisters for their love, encouragement and guidance. Special gratitude to my friends; Sharon, Eva, Ray & Roger for their continuous support during this journey. Always and Forever!

Sylvia Charlene Karanja

An expression of gratitude to my family and friends for their support throughout my studies. Additionally, I would like to give a special thanks to program coordinator for the Bachelor's program in informatics, Viktoria Svenby. Her guidance through the application process for courses in informatics, both at Lund's university and other universities, was invaluable for my process of becoming eligible for the Master's program in Information Systems.

Björn Arvesen

Content

1	Introduction.....	1
1.1	Background.....	1
1.2	Problem Area and Research Question	2
1.3	Purpose	3
1.4	Delimitation	3
1.5	Definitions	4
2	Theoretical Background.....	6
2.1	Diffusion of Innovation Theory.....	6
2.2	The Process Perspective	7
2.3	Understanding Developers	9
2.3.1	Third-party Developer Roles.....	9
2.3.2	Third-party Developer Motivations for Platform Adoption.....	10
2.4	Platform Design	11
2.4.1	Boundary Resources.....	12
2.4.2	Critical Mass	13
2.4.3	Organizational Identity and Differentiation	14
2.5	Governance and Control	15
2.5.1	Teachings from OSS	15
2.5.2	Platform Rules.....	16
2.5.3	Balancing Creative Freedom and Control.....	17
2.6	Research Framework	18
3	Methodology	20
3.1	Research Strategy	20
3.2	Data Collection	20
3.2.1	Argument for Case / Informant Selection	21
3.2.2	Interview Guide Design	23
3.2.3	Interviews	24
3.3	Data Analysis.....	25
3.4	Research Quality.....	28
3.4.1	Reliability & Validity.....	28
3.4.2	Ethics.....	29
4	Empirical Results	30
4.1	Understanding Developers	30
4.1.1	Developer Motivations in Platform Adoption.....	30

4.2	Platform Design Strategies	34
4.2.1	Boundary Resources.....	34
4.2.2	Critical Mass	36
4.2.3	Organizational Identity and Differentiation	37
4.3	Governance and Control	39
4.3.1	Platform Rules.....	39
4.3.2	Balancing Creative Freedom and Control.....	40
4.3.3	Communication Structures and Leadership	41
5	Discussion	43
5.1	Developer Motivations for Platform Adoption	43
5.2	Platform Design	45
5.2.1	Boundary Resources.....	45
5.2.2	Critical Mass	46
5.2.3	Organizational Identity and Differentiation	47
5.3	Governance and Control	48
5.3.1	Platform Rules and Communication Structures	48
5.3.2	Balancing Creative Freedom and Control.....	49
5.4	Enriched Research Framework.....	50
6	Conclusion	54
6.1	Research Question and Purpose	54
6.2	Key Findings.....	55
6.3	Implications for Future Research	56
	Appendices	57
	Appendix 1: Interview Guide	57
	Appendix 2: Interview Transcript [Rsp1].....	59
	Appendix 3: Interview Transcript [Rsp2].....	70
	Appendix 4: Interview Transcript [Rsp3].....	76
	Appendix 5: Interview Transcript [Rsp4].....	82
	Appendix 6: Interview Transcript [Rsp5].....	87
	Appendix 7: Interview Transcript [Rsp6].....	97
	References	106

Figures

Figure 2.1: Key components of a platform ecosystem by Schuermans, Vakulenko and Constantinou (2015).....	8
Figure 2.2: A Conceptual model for the process of platform establishment.....	8
Figure 2.3: Boundary resources model from Ghazawneh & Henfridsson (2013).....	12
Figure 3.1: Kvale and Brinkmann's (2009) seven stages of interview research.....	21
Figure 3.2: Snapshot of coding summary from NVivo.....	27

Tables

Table 2.1: Summarized research framework.....	18
Table 3.1: Summary of interview details	25
Table 3.2: Thematic codes	26
Table 5.1: Summary of key factors for consideration in platform establishment	50

1 Introduction

This chapter outlines the background and motivations of the study with regard to the platform economy and third-party development. It clarifies the problem area and lists the limitations of the study. Additionally, the 'definitions' section facilitates understanding of key concepts that are somewhat specific for this research topic.

1.1 Background

An increasing number of organizations are deciding to extend their business model by moving to platform-based environments. Here, core products are complemented and extended by external innovators known as third-party developers (Manner, Nienaber, Schermann, & Krmar, 2013). Platform-based software ecosystems, unlike traditional software development, tap into the expertise of diverse developer communities to create superior products (Reuve, Sørensen, & Basole, 2017). This concept, also known as platform thinking, builds on the competitive value gained from new ideas provided by outsiders (Boudreau & Lakhani, 2009; Chesbrough, 2007). It creates a broader innovation ecosystem that turns suppliers and competitors into complementors and partners, in what is referred to as a “new type of scope economics” (Leijon, Svenheden, & Svahn, 2017). The success is demonstrated Apple’s developer program, which has been able to drive extraordinary user demand by extending the value proposition of core devices through millions of applications. This representative of mobile ecosystems still remains the “dominant design” of successful platform business models (Gawer & Cusumano, 2014). The vast network of third-party developers has propagated self-enhancing mechanisms which have increased the firm's competitive sustainability (Wan, Cenamor, Parker, & Alstyne, 2017). Thus, Apple has provided a benchmark for upcoming strategies of external innovation (Ghazawneh & Henfridsson, 2013; Rickmann, Wenzel, & Fischbach, 2014).

It is evident that the exploitation of vast networks of third-party developers has allowed the big actors in mobile operating systems such as Google’s Android and Apple’s iOS to gain unparalleled market share (Eaton, Elaluf-Calderwood, Sørensen, & Yoo, 2015; David S. Evans, 2016). However, third-party development has evolved beyond individual mobile-app development and is now being diffused into business to business (B2B) settings (Schuermans, Vakulenko, & Constantinou, 2015). Due to the differences in properties and platform access, there was suspicion that the approach to developer attraction and retainment would change (Parker, Alstyne, & Jiang, 2017). But targeting individual developers is argued to still be the most viable strategy, regardless if the developer acts on his own or is clustered in an organization that operates in B2B settings (O’Grady, 2013). According to Schuermans, Wilcox, Hecht, and Voskoglou (2017) and Dr. Prashant Pradhan, Chief Developer Advocate IBM India, organizational development-decisions such as choice of software platform tools and programming languages still reside with developers rather than the c-suite managers like CIOs or CTOs (Sarkhel, 2017). The argument posed is that these managers have neither the time nor insight to engage in the direct work of frontline developers, and often seek advice and endorsements in such decisions. Preferences of individual developers have thus become increasingly important,

as they begin to understand their value and expect more from the platforms on which they choose to innovate (Schuermans et al., 2017). As platforms and technology evolve, developer

The ability to attract and retain developers relies on building an optimal developer environment (O'Grady, 2013). This implies a transition from in-house R&D to having the business revolve around a platform where externals are supported in their co-creation and design-efforts. Thus, an organization simply becomes a platform company once it opens up its core product to external innovation (Boudreau & Lakhani, 2009). Some of the most promising start-ups such as Airbnb and Uber, together with Google, Amazon, Apple and Microsoft which were ranked as the most valuable companies of 2016, all use platform business models (Gandel, 2016). These models drive value through direct or indirect network effects which create value by facilitating on-demand exchange between groups of consumers and producers (Staykova & Damsgaard, 2015). There is a nexus of rules and architectures that must be open to allow for third-party development but governed and maintained such that participants abide by the set rules (Wan et al., 2017). While this model has existed since the age of newspapers connecting subscribers to advertisers, the current age of technology and especially cloud infrastructure has made the implementations of software platforms cheaper, more scalable and simple (Smedlund & Faghankhani, 2015). New technologies do however come with challenges of their own, and the rackful complexity of platforms is exemplified by endeavours that have failed miserably, such as like eBay's 'Billpoint' or Google's 'Health' platforms (Alstyn, Parker, & Choudary, 2016).

Some companies start off with the platform business model, but most are "closed" product companies that later transition, as in the case of Amazon which launched as a retailer in 1994 only to transition to the Amazon Marketplace six years later (Zhu & Furr, 2016). A key trigger of such transition has been the changes in consumer behaviour in terms of expecting more in service, support, capabilities and most importantly a customized experience (Hobcraft, 2017). It is evident that the platform business models span over a variety of industries, ranging from social networking to payment technologies, and even health and fitness most of which are highly interrelated as profits flow from multiple players (Evans & Gawer, 2016). These companies however differ in how they implement platform business models and manage third-party development (Kude, Dibbern, & Heinzl, 2012).

The success of actors in mobile industry and e-commerce has shown the vast amount of value to be captured from third-party development (Gandel, 2016). As business models based on platforms and ecosystems are now being diffused over various settings, organizations begin to acknowledge the influence and importance of individual developers (O'Grady, 2013). Attracting and retaining developers is seemingly going beyond means of monetization, SDKs and APIs, where new considerations need to be made at every stage of platform establishment (Tura, Kutvonen, & Ritala, 2017). Establishing and maintaining a healthy developer ecosystem can impact various aspects of the business as in accordance with the arguments and configurational dynamics stated (Altman & Tripsas, 2014). Thus, it has come to be of great importance to understand what to take into consideration when aiming to establish a platform ecosystem that attracts and retains developers.

1.2 Problem Area and Research Question

In order to achieve new growth through co-creation and collaboration, organizations in digital businesses need to appeal to the needs of developers as the primary drivers of platform adoption.

There has been several studies on the importance of network effects (Eisenmann, Parker, & Alstynne, 2006), theoretical benefits of developing a platform ecosystem (Hagiu & Rothman, 2016; Top, Dilek, & Çolakoglu, 2011), and governance of third-party development through the use of boundary resources (Ghazawneh & Henfridsson, 2010), but very few focus on the factors that have to be considered to build environments that both attract and retain third-party developers. Parker et al. (2017) argue that more developers give a platform far more chance of success, by causing a shift that could positively impact growth. Upcoming companies and those that wish to use third-party developers to extend their digital business models thus need to cultivate an ecosystem designed in accordance with the needs of the targeted actors (Alstynne et al., 2016). This is also motivated by the fact that even the most valuable companies such as Amazon started out as product companies before making the leap to platform, even though such a success story is far from guaranteed (Zhu & Furr, 2016). We therefore see a gap in theory regarding a comprehensive view of what factors such companies can consider in this process and hence formulate the following research question;

Which factors are important in a platform ecosystem establishment - that attracts and retains third-party developers?

This will be analysed by examining two primary aspects:

- 1) The process of platform establishment and the corresponding key factors for consideration.
- 2) How the identified factors affect platform diffusion among developers.

1.3 Purpose

The purpose of this study is to develop a comprehensive list of factors that c-suite technical managers can consider when establishing a platform innovation model marked by an external developer ecosystem. This is based on the properties of a digital business model that attracts and retains third-party development and the motivation exhibited by Zhu and Iansiti (2012) empirical model of how platform entrants can overcome incumbents based on the strength of developers' indirect network effects. Leveraging the benefits of innovation has recently required strategies for "platform evangelism" in managing third-party contributions and this study aims to fill a void in the limited insights that currently exist on how firms can implement this shift (Parker et al., 2017). Similarly, supported by the diffusion of innovations theory (Rogers, 2010), platform boundary resources model (Ghazawneh & Henfridsson, 2010) as well as the more process-based studies of platform design, launch and governance (Tura et al., 2017), this study aims to outline and explain key considerations for establishing a platform ecosystem with a high diffusion rate that supports, captures and maintains developer attention for competitive sustainability.

1.4 Delimitation

This study is delimited to how firms can build and extend digital business models centered on the role of external innovators. External innovators in this case refer to third-party developers or third-party development in itself considering both B2B and B2C contexts. Here, they could

be in the form of customers, product extenders, data harvesters or resellers and distributors (Vakulenko, 2016). Given the perceived dominance of mobile ecosystems, there still lacks adequate literature that reliably classifies platform companies. The speculated reason is that even with the common denominator of generating network effects, the method with which such companies create, capture, deliver and protect their value still differs (Evans & Gawer, 2016). This is especially due to the emergence of cloud infrastructure, online services and data-centric solutions that have greatly diversified the types of business models emerging and also the overlapping properties that exist in the ecosystem (Schuermans et al., 2015). A good example is in Microsoft, Apple, Google and Amazon which as the key role models of successful platform ecosystems all have different digital business models and continue to innovate while offering unique value propositions (Basole, 2009). For these reasons, this study explains the considerations that an organization has to evaluate in order to develop a third-party developer-centric product, irrespective of the type of business model. To note is that this research does not look at outsourcing IT services, but rather at extending a company's core product digitally to reach new markets through third-party development.

1.5 Definitions

Platforms

This term is in this study synonymous to digital or software platforms which imply a set of interrelated technology layers that form a common resource base, from which derivative software can be developed and integrated. It implies a shift of design capabilities to external actors, who then make their development of applications compatible with the core platform (Ghazawneh & Henfridsson, 2010).

Ecosystems

Related to platforms and borrowed from the biological “ecosystem”, this term refers to a set of entities or organizations highly co-dependent on each other’s input and output (Moore, 1996). It is a web of collaborating and competing firms that offer connected products and services (Iansiti & Levien, 2004).

Platform owners or hosts

Are defined as the organizations or individuals that host the platform from which third-party developers can integrate with the core products. This is usually done through available Application Programming Interfaces (APIs).

Digital business models

This implies the use of technology to create new value for a company. The term applies to both technology companies and traditional asset-heavy players looking to transform their businesses digitally (Weill & Woerner, 2013).

Third-party developers

Third-party developers are also referred to as complementors, external innovators or integrators in this study. According to Gawer and Cusumano (2002) these actors are defined as those building complementary services or products around a core device. This term is also used interchangeably with simply 'developers' in this research paper.

Permission-less versus negotiated platform access

Due to the different nature in the types of businesses that use third parties, the relationships between hosts and developers will also be quite different. In permission-less contexts, there is less control of the often-high number of developers who build on top of the platform. In such instances, platform hosts use terms and conditions easily distributed and shared with new integrators. Negotiated platform access on the other hand implies arm's length relationships where third-party developers and partners are vetted and maintained by formal contractual agreements in direct platform service sales (Parker et al., 2017).

Developer Marketing

This is the art of attracting and engaging developers much like consumer marketing. It is a challenge that is prevalent in today's software platform ecosystems where such companies devise strategies of getting developers to work on their product.

2 Theoretical Background

This chapter outlines the key supporting literature that form the foundational basis of this study. Paragraph 2.1 and 2.2 describes the core theories and justifies thematization, as well as the theoretical basis for later analysis of empirical findings. The paragraphs that follow highlights relevant theories in IS and innovation theory relating to the core themes. Lastly, the chapter compiles themes, subthemes and literature into a theoretical framework that serves as the basis for the data collection and discussion of results in chapter four and five.

2.1 Diffusion of Innovation Theory

This study finds its basis in diffusion of innovation theory (DOI) (Rogers, 2010), framed in a process perspective similar to the studies by (Ghazawneh & Henfridsson, 2010) and (Tura et al., 2017). DOI theory explains the rate in which new technologies will be diffused and groups prospective adopters based on what phase they are expected to adopt the innovation (Rogers, 2010). This research theory builds on the assumption that upon learning the advantages of an innovation and how it triumphs current solutions, prospective users are more likely to adopt (Bui, 2015). The aspect of grouping adopters is outside the scope of this paper, but the five factors impacting diffusion make an appropriate tool for later analysis. These are listed as;

- **Relative advantage:** Is defined as the degree to which an innovation is perceived better than the tools and solution it supersedes (Rogers, 2010, p. 6).
- **Compatibility:** Is the degree to which the new innovation is perceived consistent with existing values, experiences and preferences of potential adopters (Rogers, 2010, p. 6).
- **Trialability:** Also known as testability, is the degree to which an innovation may be experimented on a limited basis (Rogers, 2010, p. 7).
- **Observability:** This is the degree to which results of an innovation are visible to others (Rogers, 2010, p. 7).
- **Complexity:** Is the degree to which an innovation is perceived as relatively difficult to understand and use (Rogers, 2010, p. 7).

The first four factors have a positive correlation with platform adoption, meanwhile the last one has a negative correlation (Rogers, 2010). When developers decide whether to adopt a specific platform or not, they will consider the quality of the bundled services and technologies that are available. In relation to diffusion of innovation theory, developers thus represent the prospective adopters of technology, meanwhile the different properties and resources of platforms represents the new innovations that will be either adopted or discarded. Innovation is here defined as “...a new product/service that is developed in the environment of a platform ecosystem” (Shuradze, Wagner, & Wagner, 2015, p. 2). From a host’s perspective, this implies that considerations of the factors defined by Rogers (2010) will impact the platform’s diffusion among developers.

Furthermore, the study is supported by innovation networks literature (Boland, Lyytinen, & Yoo, 2007; Chesbrough, Vanhaverbeke, & West, 2006; Van de Ven, Polley, Garud, & Venkatraman, 2008; Yoo, Lyytinen, & Boland, 2008, 2009) and boundary objects theory

(Bergman, Lyytinen, & Mark, 2007) which demonstrate a delicate problem in control and coordination of knowledge resources characterized by multiple heterogeneous actors in an ecosystem (Boland et al., 2007). Similarly, this research takes on third-party developers and their motivations as key actors of an innovation network and from the perspective of platform owners outlines the key considerations in proper design and governance of such an ecosystem to foster co-creation. This acknowledges Youngjin, Ola, and Kalle (2010) and Star and Griesemer (1989) argument that digital innovation gives rise to a new type of product architecture and changes on how firms organize for innovation.

2.2 The Process Perspective

Implementing a successful digital business strategy requires a clear plan to ensure maximization of the benefits gained from technology focused initiatives. In the context of platform thinking and use of third-party development, several studies have been identified that provide a benchmark on the main factors of a specific aspect of platform establishment (Schreieck, Wiesche, & Krcmar, 2016; Tiwana, 2014, 2015). Research by Tura et al. (2017) show in their framework of platform design how different factors are considered in different phases of platform roll-out. Many of these have overlapping properties, which indicate that factors considered early in the process influences the later stages. In this process, they cite platform architecture, value creation logic, governance and platform competition as essential in the “*scheme for creating conditions for sustaining a multi-actor platform-based business*” (Tura et al., 2017, p. 2). Tiwana, Konsynski, and Bush (2010) do in their framework for understanding platform-based ecosystems discuss six aspects, consisting of the use of platform-centric approaches for competitive advantage, the use of boundary resources and technical architecture to harness outside expertise, platform governance with regard to balancing control and finally the impact of internal and external environments. Consequently, this highlights Benzell, Lagarda, and Alstynne (2017) study on the impact of APIs in a firm’s performance that justifies Ghazawneh and Henfridsson (2010) process perspective on governing third-party development anchored on boundary resources design. The key components of a platform ecosystem can therefore be summarized as in Figure 2.1 by Schuermans et al. (2015) where people (developers) are the core actors in the use of a technology platform governed by rules of interaction:

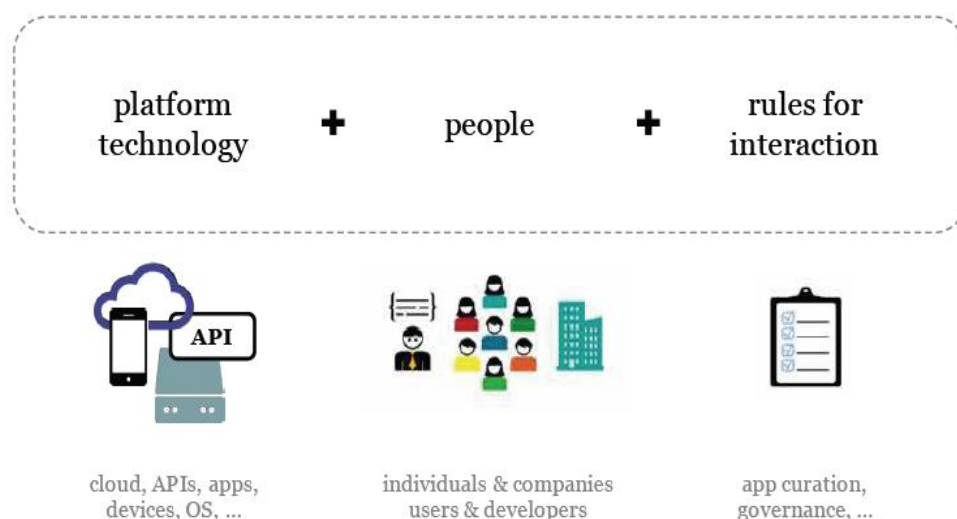


Figure 2.1: Key components of a platform ecosystem by Schuermans, Vakulenko and Constantinou (2015)

Considering all the factors identified in the sample studies above, we adopt an interlinked or iterative process perspective consisting of three key components: (1) Understanding third-party developers as the key actors in building platform strategies either through apps or services Schuermans et al. (2015). (2) Platform design where organization make decisions on how to best realize developer preferences, and (3) Governance and control on how to ensure stability and alignment once the platform is up and running (Goldbach & Kemper, 2014; Lin, Li, & Whinston, 2011; Schrieck et al., 2016) In each of these components, as will be presented in the following sections, we present an inclusive logical flow summarized from the above studies on platform ecosystem frameworks (Figure 2.2). We first review developer motivations from which arises the effective design of platform boundary resources as an interface between developers and core product (Ghazawneh & Henfridsson, 2013). Having the technical infrastructure set in place requires having more users of the system in order to drive network effects on the platform (Hagiu & Rothman, 2016). This however impacts the organizational identity and differentiation according to (Altman & Tripsas, 2014). For this reason, proper design rules and communication structures need to be established given the new dynamics of external developers while maintaining balance in the degree of platform openness for sustainable competitive advantage (Laffan, 2011; Schrieck et al., 2016). These key factors thus form the main themes and subthemes chosen for discussion in this chapter.



Figure 2.2: A Conceptual model for the process of platform establishment

2.3 Understanding Developers

The first and foundational phase of platform establishment revolves around understanding your key actors (Tura et al., 2017). Described as the new ‘kingmakers’, developers are the key actors used to create, deliver and capture value for platform businesses meanwhile increasing demand for products and erecting barriers to competition (Bender & Gronau, 2017; O’Grady, 2013). This is motivated by the fact that platforms have a main goal of developing new capabilities beyond the original core product portfolio (Goldbach & Benlian, 2015a). When assessing different value propositions from platform hosts, developers represent prospective adopters of technology innovations, as described by Rogers (2010). In order to create an optimal value proposition, hosts thus have to understand how relative advantage, observability, testability, compatibility and complexity translates into actionable preferences of developers.

Creating a fruitful platform ecosystem does however go beyond attracting developers, since they also need to be incentivized to stay once they have entered. Goldbach and Benlian (2015b) refer to this as ‘platform stickiness’, which is important for the platform’s long-term viability and success. Bender and Gronau (2017) underline that the real value of a platform is derived from the innovations third-parties contribute to the ecosystem. Thus, the innovations developers bring needs to remain in the ecosystem in order to ultimately create a one-stop shop where all functionalities that any developer, partner, host or end-user could desire are available.

2.3.1 Third-party Developer Roles

Despite the role of internal developers in innovating and improving a company’s product, external developers can also be beneficial in the following ways (Vakulenko, 2016):

- *As customers:* Companies can earn revenue by providing premium services and tools to developers. Twilio is one such company that provides telephony and SMS services that developers can integrate into their apps (Twilio, 2018).
- *As resellers and distributors:* Developers can integrate a company’s product into their own applications, hence attracting and delivering to customers in new markets. The use of developers as a sales channel is also not limited to internet companies as seen in Walgreens as a retailer that through its Photo Prints API has allowed users of mobile apps in over 8000 Walgreens locations in the US to print photos through a variety of photo apps (Walgreens, 2018). Uber’s affiliate program also allows developers to integrate on-demand transportation services into their apps and services (Uber, 2018).
- *As product extenders:* Addition of new features to the core product spanning needs not originally anticipated increases the value of a company's product and brings new users on board. Case in point is DJI which allows developers to create drone-based applications for tourism, disaster response etc. (DJI, 2018).
- *As data harvesters:* Advertising companies like LinkedIn and Facebook through their login APIs can collect data and intelligence about users in order to create more targeted products and advance their core businesses (Vakulenko, 2016).

2.3.2 *Third-party Developer Motivations for Platform Adoption*

Third-party development as defined by Ghazawneh and Henfridsson (2013, p. 2) is a type of systems development where the third-party developer, on behalf of the platform owner, develops applications, services or systems for satisfying end-users of the platform often with contractually oriented relationships (Boudreau & Lakhani, 2009). Just as other companies seek to attract consumers in consumer marketing, platforms need to attract developers in what can be called ‘developer marketing’ (Schuermans et al., 2015). A platform’s choice of business model provides a basis of how the developers create, capture or deliver value within an ecosystem (Vannieuwenborg, Mainil, Verbrugge, Pickavet, & Colle, 2012) and determines the cumulative commitment of the developer based on the value proposition presented by the platform (Hsieh & Hsieh, 2013). Understanding the motivations of these outside innovators therefore is essential in building platforms that attract and retain them in order to propagate eventual participation of end users (Parker et al., 2017). According to Accenture’s 2018 developer ecosystem survey, developers acknowledge the gap in platforms’ ability to meet their needs imploring them to build healthy ecosystems that do better in educating, supporting, inspiring and engaging them (Accenture, 2018). However, formulating optimal business models to cater for both the developer’s and platform provider’s needs is not always clear. But key elements to consider to deliver optimal developer-value proposition include types of promotion channels, key partners and providers, resources available, associated support activities and cost structure (Vannieuwenborg et al., 2012).

Traditional developer motivations in open innovation projects include having fun, reciprocity, fairness, transparency and full access to code (Shah, 2006). Despite Shah (2006) study being primarily based on collaborative environments such as OSS development, there are studies showing that monetary rewards in OSS developments can actually increase the project’s reputation and lifespan (Midha & Bhattacharjee, 2012). Nonetheless, third-party development on proprietary platforms has more similarities to competitive markets than collaborative projects (Boudreau & Lakhani, 2009). In such contexts, developers are primarily motivated by monetary incentives or pricing models (Hsieh & Hsieh, 2013). Similarly, Kankanhalli, Ye, and Teo (2015) shows in his study that anticipated extrinsic rewards like monetization have a positive correlation with third-parties intentions to innovate, which applies especially for developers who have not yet entered the ecosystem.

Vannieuwenborg et al. (2012) emphasize third-party developers’ need of business models to capture value from their applications. Despite the fact that these have seen increased diversification during the last years, it is evident that payment models are often deprioritized by developers, who do not want to get bogged down in transactional systems and processes (Manner et al., 2013). Nonetheless, a platform’s opportunities for monetary rewards has a direct correlation to developer’s commitment, meaning that platforms that facilitate this process will have an easier time attracting and retaining third-party developers (Hsieh & Hsieh, 2013; Oh, Koh, & Raghunathan, 2015).

However, Accenture (2018) reports the pragmatic nature of developers preferences and states that the need to improve or gain new skills tops monetization in what they want from most ecosystems. This is followed by accurate and up-to-date content, product support, openness, market position, how well it is integrated with other companies and finally how the platform is differentiated from others. In his study of 750 US based professional developers, Free (2018a) finds that the traditional efforts of platform providers using events and giveaways are less at-

tractive. Instead, these developers ranked the factors they value most as; platform market position (90%), technically accurate content (82%), on time support (81%), improving and gaining new skills (78%), integration with other leading platforms (76%) and revenue opportunities (64%) (Free, 2018; Oh et al., 2015). Companies able to leverage these preferences could benefit from locking in developers and preventing them from switching to other platforms. As is, what motivates developers to switch platforms is a focus on latest technologies, future looking aspects, ease of use and potential for career growth (Free, 2018; Vannieuwenborg et al., 2012).

Given this diversity in motivation, companies need to consider carefully whether they need to focus more on a collaborative community or a competitive market as it will affect the type of external innovators who participate (Boudreau & Lakhani, 2009). According to these authors, communities require mechanisms of knowledge exchange and interactions to give its members a sense of identity and affiliation, contrary to markets which require formal competitive mechanisms that may discourage knowledge sharing. Observing the different effects of such control modes thus gives insight into the work-related performance and loyalty of developers to platforms (Goldbach & Benlian, 2015a).

2.4 Platform Design

When motivations have been investigated, companies enter the second phase by designing a platform that aims to cater to the motivations of the key actors (Tura et al., 2017). Developing a successful platform strategy is no easy or well-defined task. Gawer and Cusumano (2002) suggest 4 dimensions for consideration in successful establishment;

- **Scope:** This has to do with what is done in-house versus what is left to third-party complementors. This often creates a push and pull with regard to which architectural details to disclose, revenue sharing and resource ownership strategies (Popp & Meyer, 2010). The aim is to ensure integrity and niche preservation in the organization while fostering external innovation.
- **Product technology:** How a company builds the architecture of their product considering foundational technology relative to that of surrounding interfaces impacts the adoption levels of the platform by complementors (Schilling, 2009).
- **Relationships with external complementors:** Building a community of third-party developers is one thing but maintaining their loyalty and keeping them engaged in such a way as to increase their switching costs is another. Standards need to be set on how complementors relate with each other and with the core platform to ensure continuous and mutually beneficial value creation (Gawer & Cusumano, 2002).
- **Internal organisation:** Since platform ecosystems create co-opetition where organizations collaborate and compete at the same time, conflicts of interest between internal and external partners should be mitigated well in advance (Bengtsson & Kock, 2000).

This chapter and the next highlights some of the key components in the design, implementation and governance of a prosperous platform ecosystem encapsulated in three determinants of robustness (ability to survive disruptions), productivity (efficiency in converting inputs to outputs) and niche creation (meaningful diversity) (Iansiti & Levien, 2004).

2.4.1 Boundary Resources

In platform-based businesses, strategy has changed from developing complements to providing third parties with assets to enhance company core products (David S Evans, Hagiu, & Schmalensee, 2006). This implies that design capabilities have had to transfer from internal to external actors (Von Hippel & Katz, 2002). To support these actors, platform owners at the very basic provide technological resources that usually consist of tools such as software development kit (SDK) and a number of related APIs and regulations (Youngjin et al., 2010). These are resources aimed to support the development work of the third-party developers. With these, developers get access to core modules and the possibility to add value to the core platform through applications building (Ghazawneh & Henfridsson, 2013). As resources are provided, measures must be taken to prevent development of applications that risk infringing the platform (Benzell et al., 2017). Resources provided are thus “boundary” in the sense that they provide the tools and regulations required to govern the relationship between the platform owners and application developer (Bianco, Myllarniemi, Komssi, & Raatikainen, 2014).

Over time however, this level of boundary resources has not been enough to capture and retain developer attention, hence the emergence of new boundary resources otherwise known as ‘social boundary resources’ (Ghazawneh & Henfridsson, 2013). These are often generated as a result of how the platform owner perceives external contribution or as a result of how the third-party developers use the resources (Helm, Holland, & Gangopadhyay., 1990). The process of boundary resources design according to Ghazawneh and Henfridsson (2010) is therefore a continuous task for the platform owners as seen in Figure 2.3 below, aimed at adapting to the dynamics of the third-party development community.

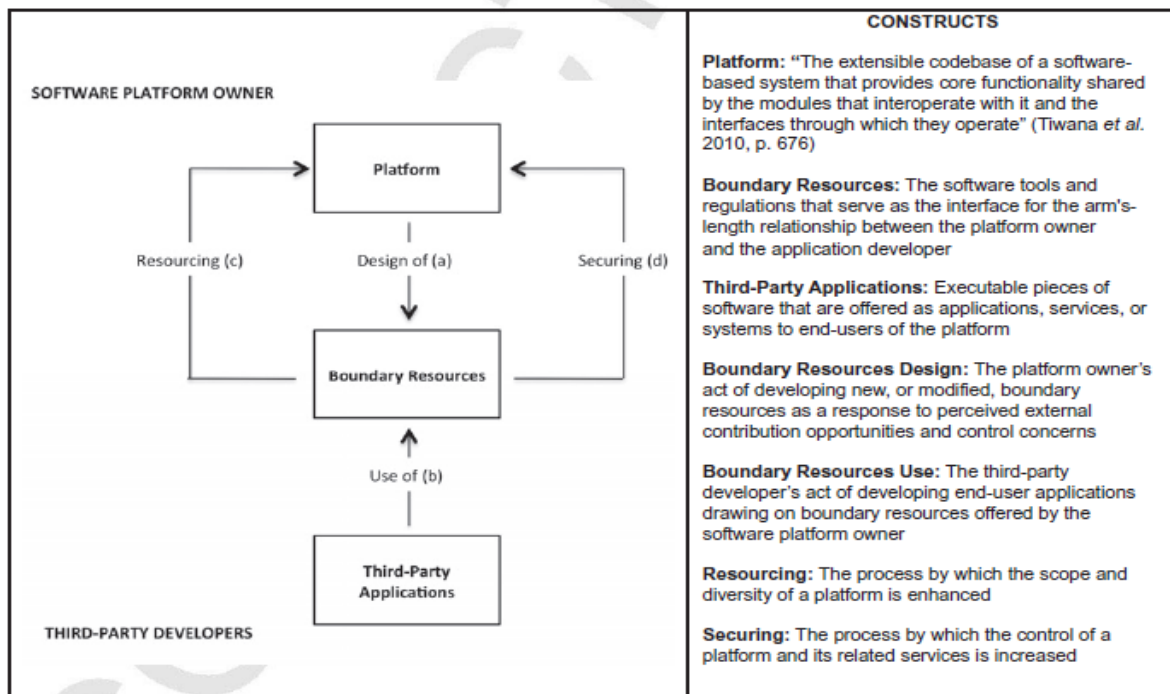


Figure 2.3: Boundary resources model from Ghazawneh & Henfridsson (2013)

Boundary resources design involves enhancing the diversity and scope of the platform in terms of knowledge or capabilities (resourcing) together with control through modifying developer agreements or in application review processes (securing) (Andrew H Van de Ven, 2005). Other

ways to do this is by optimizing the platform's distribution channel regarding the reach to a wider user community (Ghazawneh & Henfridsson, 2010). This today is mostly done through cloud services that allow real time integrations, as opposed to the mainstream App Store channels owned by mobile ecosystems. Similarly, the need to protect a platform ecosystem from infringements by other competitors is paramount. An example given by Ghazawneh and Henfridsson (2010) is of Apple's move to block meta/cross platforms such as Adobe's Flash Player in their Safari web browser by updating SDK agreements and rules. This was indeed a delicate issue that left third-party developers with a lot of concerns, especially since some of their applications required this functionality to run on Apple's devices. Another aspect emphasized by Nylén and Holmström (2015) is the importance of building usable interfaces taking into considerations aesthetic properties that evoke engagement. This involves creating user experiences that digitally invoke clear meaning and value to the third-party users (Diller, Shedroff, & Rhea, 2005).

Platform companies should therefore consider the design and evolution of their boundary resources, also known as self-resourcing, either at a technological level or social level in order to respond to any limitations in the existing boundary resources model (Ghazawneh & Henfridsson, 2013). This is supported by Mohagheghzadeh and Svahn (2016) who note that there are often discrepancies in available platform resources and what is actually perceived by developers. Thus, healthy ecosystems are shaped by hosts constantly tuning boundary resources according to the preferences, feedback and actions of developers. Despite actively working with evolving boundary resources, hosts must anticipate and accept a partial loss of control as the platform scales, and nurture communicative relationships with developers (Eaton et al., 2015).

2.4.2 *Critical Mass*

Platform businesses which rely on direct and indirect network effects to attract both creators and end-consumers tend to face issues with demand at launch (David S Evans & Schmalensee, 2010). David S Evans (2009) uses analogies of chemical reactions to describe the environment required for the platform-based business model to succeed. Just like a chemical reaction, a platform-based business model needs an igniting catalyst to be able to accelerate effect. The 'effect' is in this case represented by reaching a 'critical mass' of users (David S Evans & Schmalensee, 2010). When both sides of platform's users, being represented by creators and consumers, have reached a sufficient number, the platform has become liquid enough to maintain sustainable growth (David S Evans, 2009). If not, the platform will quickly implode as users move to healthier ecosystems. As can be seen before the acquisition, Google's attempt to compete with YouTube through 'Google Video' failed tremendously due to not being able to generate enough content to attract viewers and not enough viewers to attract content-creators (Fortt, 2006).

There are different ways to ignite critical mass, which bears similarities with those of new product launches. Hart and Tzokas (2000) compile some of the most prominent strategies to include logistics, advertising, marketing and pricing. For an example, companies may need to be ready to initially slash prices to generate user attention, which requires enough financial liquidity to sustain losses for some time after launch (Altman & Tripsas, 2014). The nature of the platform business does also need to be taken into consideration when formulating a strategy to reach critical mass. The platform-host needs to understand in what way users are going to enter the platform environment. David S Evans (2009) gives three examples of user adoption of a platform:

- *Sequential entry.* It is possible to get one group of entry on board the platform before the other one. An example is platforms whose business model builds on advertisement-supported content. Since viewers only come for content and do not care about the advertisement, the platform can target viewers first and turn to the profit-generating advertisers when a critical mass of viewers has been reached. A common strategy is also when platforms offer products with trial versions and basic versions for beginners then premium services for experts (Bhargava & Choudhary, 2004).
- *Entry with significant pre-commitment investment.* There are situations where users of the platform must make significant economic investments prior platform launch. Video Game developers do for example invest in creating games, without knowing the commercial success of the platform or console they intend to sell it on. To get away with this, the platform provider must have a strong brand and a track-record of previously successful launches.
- *Simultaneous entry.* This is the difficult position where most platform business models start. Both creators and consumers have to join the platform at the same time, as both are crucial to the platform's value proposition. The platform needs to exploit indirect network effects to build participation on both sides incrementally.

Taking on platform incumbents thus requires uniqueness in value creation, or a widely adopted and appreciated core product that can be built upon (Tura et al., 2017). This implies focusing on a core value proposition that others are incentivized to build on. David S Evans (2009) also underlines that platform leaders need to actively seek partnerships to ensure a sustainable diffusion rate of the platform.

2.4.3 Organizational Identity and Differentiation

Markets faced with the latecomer disadvantage need to employ differentiation in their product and provide an ecosystem that offers unique services to attract third-party need (Wan et al., 2017). This can be through improvement of a platform's internal features to provide superior value or by offering services to a variety of user types to cater to different preferences to a platform's characteristic (Sordi, Nelson, Meireles, & Silveira, 2016). But even as companies differentiate themselves from competitors, they should not lose sight of their identity. A company's primary attraction always remains on its core product despite third-party value additions (Zhu & Furr, 2016). This especially in hybrid business models enables them to retain their core customer base by increasing product value while at the same time source for new external opportunities in digital business models.

Altman and Tripsas (2014) discuss the organizational capabilities required to make the transition to a multi-sided platform business model. Organizational capabilities referring to the daily routines and processes that together with input flows comprise the resources that differentiate an organization from competitors (Winter, 2003). One of the basic capabilities brought up by Altman and Tripsas (2014) is the organizational identity, which answers the question "How do we define what business we are in?". Moving from a product to a platform-business model is a considerable change and managing the organizational identity will provide the flexibility needed for a successful transition (Altman & Tripsas, 2014). At this point the platform should be differentiated through enhanced value in its intrinsic characteristics in order to offer high quality products (Wan et al., 2017).

Additionally, organizations need to reflect on their role in the ecosystem they are about to create or enter. Becoming the platform leader means having ownership of the core product upon which other companies build their products but may imply a heavy dependence on the success of its complementors (Gawer & Cusumano, 2014). A firm should therefore ensure that opening up to outside innovation does not violate their identity or integrity and expectations outsiders have on their market category (Benner, 2007). In the case of Apple, despite having to adjust strategy, resources and external relationships to accommodate the new set of ecosystem players, they still maintain the core business model of hardware manufacturing and mobile operating systems (Pisano & Verganti, 2008). Conflicts arise when platform leaders decide to compete with their complementors, fail to follow on their commitments or force complementors into large investment risks in uncertain markets (Gawer & Cusumano, 2014). Becoming a platform host does not only imply changes to the business model, but also the organizational identity, values and priorities need to be consciously managed for the transition to succeed (Altman & Tripsas, 2014).

2.5 Governance and Control

When the platform has been designed and launched in accordance with the preferences of key actors, governance and control aspects come in (Tura et al., 2017). To be able to truly reap the rewards of third-party development, the host needs to support the ecosystem and ensure it is evolving in the preferred direction (Gawer & Cusumano, 2002). This means implementing, maintaining and updating different means of control on the platform (Tiwana et al., 2010).

2.5.1 *Teachings from OSS*

Using third-parties for external innovation is not a new concept, as it has been present in the form of open source software (OSS) development. OSS development implies that a community of external developers voluntarily works on a software development project together (Feller & Fitzgerald, 2002). OSS projects were initially built on the vision of equal creative freedom and unanimous acceptance of additions and changes. However, direct democracy tends not to scale well and halts performance as the projects grows bigger. O'Mahony and Ferraro (2007) note that successful governance stems from the community's agreement of a basis of authority. The immense success of open source projects such as the operating system Linux is for an example often attributed to the coordination by founder and community leader Linus Torvalds (H. W. Chesbrough & Appleyard, 2007). It is however not the leader per se, but the communication structures brought by leadership that are essential for OSS success (Hemetsberger & Reinhardt, 2004). Shah (2006) claims there to be two primary types of contributors to OSS communities, those driven by the need for improvement and those driven by the fun of co-creation. Those driven by need for improvement are especially sensitive towards the community becoming more "gated", with restricted access to source code and platform-ownership being primary points of concern. To maintain control meanwhile satisfying both groups, OSS projects have relied on lean communication in the form of forums and mailing lists (Hemetsberger & Reinhardt, 2006). It is of high importance that restrictions to the creative freedom is properly communicated to the collective.

Open source innovation networks are however characterized by a good degree of homogeneous knowledge resources and decentralized control, as opposed to the distinctively heterogeneous

nature of knowledge resources and more centralized control in platform ecosystems (Yoo et al., 2008). Nonetheless, Dellermann, Jud, and Reck (2017) note the importance of third-parties possibility to exchange information among each other in today's platform environments. Efficient communication channels between third-parties as well as with the platform host strengthens the 'social capital' of the platform and makes it more attractive (Dellermann et al., 2017). Thus, many of the insights gained from OSS development translate to the considerations needed for platform-based business models.

2.5.2 Platform Rules

When designing the platform, the host needs to communicate what rules or codes of conduct that will be used to govern the platform. If external firms are to adopt a platform as their own, the platform needs to (1) *perform a function that is essential to a broader technological system* and (2) *solve a business problem for many firms and users in the industry* (Gawer & Cusumano, 2014, p. 6). Regardless of its degree of openness, platform rules should hence be designed with strategic intent and an aim to properly manage the relationship with external innovators (Ghazawneh & Henfridsson, 2015). To succeed, actors need to take the competitive landscape into consideration (Tura et al., 2017) as platform markets are fierce, and tend to be characterized by winner-takes-all dynamics (Eisenmann et al., 2006).

The role of a platform leader implies a responsibility to support its complementors, which may have restraining implications for the strategy of the core business (Gawer & Cusumano, 2002). Hosting a platform often require an organizational shift in development processes from creativity to discipline (Altman & Tripsas, 2014). When development is conducted in-house, organizations are free in their choice and occasional replacement of tools and development interface. On platforms however, the platform host cannot freely make design decisions that will hinder or affect the work of other complementors on the platform (Staudenmayer, Tripsas, & Tucci, 2005; Tiwana et al., 2010). This includes making changes to the core product that the platform is built around. The need to evolve the platform in order to prevent it from being obsolete goes hand in hand with ensuring compatibility of complements. Ceccagnoli, Forman, Huang, and Wu (2012) underline that the foremost reason for third parties to enter an ecosystem is to signal compatibility with the platform and increase performance thanks to network effects. If compatibility is volatile, third parties are discouraged to join the platform. If a mobile manufacturer for example decides to change the screen size, whole ecosystems of complementors and application developers will be affected (Altman & Tripsas, 2014).

The rules that govern relationships with external developers regarding interaction standards in what Perrons (2009) describes as "balancing power and trust" are as well established in design in order to manage competition and collaboration within the community (Jansen, Brinkkemper, Finkelstein, & Bosch, 2009). The aspect of pricing in this context is unavoidable as it impacts the size of a customer base (McIntyre & Srinivasan, 2017). Most platforms however leverage asymmetric cross network effects until the value of the platform surpasses the price sensitivity of the paying customers (Thomas R Eisenmann, Parker, & Van Alstyne, 2010). This strategy could also attract third-party development to a platform and subsequently penetrate new markets (Iansiti & Levien, 2004). Those intending to use a platform-based business model should therefore take into consideration that it will put new constraints on the original conduct and focus of the core business (Altman & Tripsas, 2014).

2.5.3 *Balancing Creative Freedom and Control*

Control is a crucial aspect when pursuing a platform-based business model. The issue lies in governance and control of parties not under direct influence of the organization (Sagers, 2004). Gawer and Cusumano (2002) describe the actors of platform environments as leaders and complementors who are dependent on each other's success. The leader provides that core product, or the platform, and the complementor create additional products to build on the core. Both actors are characterized by platform dependency, as they need one another to gain the competitive edge. Apple's iPhone would for example not be nearly as valuable without its vast network of developers (Schuermans et al., 2015). Gawer and Cusumano (2002) argue that platform leaders must enable innovation for complementors in order to stimulate the innovation capabilities of the ecosystem. Accordingly, third-party developers comprise the complementors of today's platform-based business models (Ghazawneh & Henfridsson, 2015).

Benlian, Hilkert, and Hess (2015) discuss the challenge of ensuring an acceptable degree of "openness" in the platform to attract third-party development while at the same time being "closed" enough to maintain competitive advantage (Laffan, 2011; Tiwana, 2013). Nurturing creative freedom of developers meanwhile exercising control can be argued to be of conflicting nature (Ghazawneh & Henfridsson, 2013; Shuradze et al., 2015; Tiwana et al., 2010). Thus, the degree of openness is the level with which an external party has permission to build on or access a platform (Tiwana, 2013). Restrictions tend to have an impairing effect on innovation processes and may act as a hindering factor when aiming to attract developers to the ecosystem (Altman & Tripsas, 2014). With this lies the decision to limit or facilitate inter platform compatibility so that members of the ecosystem can interact with other ecosystem participants in what Wan et al. (2017) to as multi-homing versus single homing. A 'multi-homing' platform has an interface compatible to that of other platforms and thus has the benefit of attracting more third-party developers and hence more innovation (Boudreau, 2010). Furthermore, Hyrynsalmi, Suominen, and Mäntymäki (2016) state that in mobile platform ecosystems, three percent of developers generate eighty percent of installed applications, and the majority of these critical developers follow a multi-homing strategy. This also applies to business to business contexts, where complementors tend to follow a pluralistic strategy to minimize risk and reap innovation benefits from multiple ecosystems (Selander, Henfridsson, & Svahn, 2013). Additionally, Um and Yoo (2016) show that continuous introduction of external APIs from partnering ecosystems play a critical role in creating the diversity in functionality that is essential for platform growth.

Tiwana et al. (2010) argue that the control of resources and interfaces implies control of the platform's evolution centered on decision rights partitioning between the platform owners and external module developers. The measures become indirect, as the platform owner aims to control the premises of choice, rather than specific or individual choices. Similarly, Goldbach et al. (2015a) show that 'clan control', where developers evaluate and teach each other over loosely set premises, is the most efficient means of control that still keeps developers motivated. Especially since strict, formal control has a tendency to become tremendously time-consuming and costly (Tiwana et al., 2010). Gawer and Cusumano (2002) note that all forms of control require consensus between hosts and complementors, since platform leadership is only possible when complementors agree to comply. Thus, relinquishing enough control to drive innovation while ensuring security becomes an essential balancing act for those aiming to use third-party developers to fuel their platform-based business models.

2.6 Research Framework

In accordance with Kvale and Brinkmann (2009) description of the importance of thematizing, which guides research and provides a basis for the structuring of interview guides and questions, Table 2.1 below outlines the main themes and subthemes identified in the literature consistent with the process perspective justified above. It showcases the three main phases of considerations in extending or establishing digital business models with third-party developers in the platform economy. This is synthesized from the different frameworks, models and key concepts identified from prominent authors in the field as listed in the table and discussed above. The key considerations in each sub theme as summarized from the different authors listed is also outlined.

Table 2.1: Summarized research framework

Main theme	Sub theme	Supporting literature	Key Considerations
Understanding developers	Developer motivations	Bender and Gronau (2017) Parker et al. (2017) Hsieh and Hsieh (2013) O'Grady (2013) Vannieuwenborg et al. (2012) Schuermans et al. (2015) Kankanhalli et al. (2015)	<ul style="list-style-type: none"> ● Market position ● Support ● Cost structure and monetary rewards ● Education and gaining new skills ● Openness and control methods ● Integration with other leading platforms and strategic partnerships ● System performance
Platform Design	Boundary resources	Ghazawneh and Henfridsson (2013) Youngjin et al. (2010) Andrew H Van de Ven (2005) Eaton et al. (2015) Mohagheghzadeh and Svahn (2016)	<ul style="list-style-type: none"> ● APIs and SDKs ● Contractual agreements and other social boundary resources ● Blocking platform infringement
	Reaching critical mass	David S Evans (2009) David S Evans and Schmalensee (2010) Tiwana et al. (2010) Hart & Tzokas (2000)	<ul style="list-style-type: none"> ● Formulating a launch strategy ● Market platforms as products ● Understanding how producers and consumer will enter the platform ● Seek partnerships to enlarge the market

	Organizational identity and differentiation	Altman and Tripsas (2014) Gawer and Cusumano (2014) Pisano and Verganti (2008) Wan et al. (2017) Sordi et al. (2016)	<ul style="list-style-type: none"> • Effect on organizational identity • Strategic ecosystem alignment • Product diversification
Governance and Control	Platform Rules	Staudenmayer et al. (2005) Eisenmann et al. (2006) Tura et al. (2017) Ceccagnoli et al. (2012) Jansen et al. (2009) Ghazawneh and Henfridsson (2015)	<ul style="list-style-type: none"> • Consistency • The relationship between new innovations and core product • Developer relationship management • Analysing the competitive landscape
	Balancing creative freedom and control	Benlian et al. (2015) Tiwana et al. (2010) Shuradze et al. (2015) Goldbach and Benlian (2015) Hyrynsalmi et al. (2016) Selander et al. (2013) Um and Yoo (2016)	<ul style="list-style-type: none"> • Degree of openness • Compatibility and single homing versus multihoming approaches • Growth and control through strategic partnerships • Indirect vs direct means of control • Clan control versus formal control
	Communication structures and leadership	O'Mahony and Ferraro (2007) Hemetsberger and Reinhardt (2004) Shah (2006) Yoo et al. (2008) Dellermann et al. (2017)	<ul style="list-style-type: none"> • The properties of OSS in platform communication • Technology channels

3 Methodology

This chapter explains the rationale of the actions taken in the investigation of the research problem. It showcases the interpretive approach applied using interviews as the qualitative data collection technique. Selection of the 6 informants and cases in terms of the choice of platform companies is justified and the interview process outlined. It explains open coding strategy applied to the transcribed data collected, both manually and with the aid of NVivo software to identify the main considerations discovered as guided by the research framework above. All these is done under set ethical and professional standards that ensure acceptable quality of the instruments used, methods, and results attained.

3.1 Research Strategy

Considering the nature of our research question, and given that the theoretical ground is highly fragmented, we considered that an interpretive research strategy would be most appropriate for this study (Bhattacharjee, 2012). Often used synonymously with “qualitative research”, this approach allows us to collect data from industry experts and inductively build our theory on how organizations can create healthy platform ecosystems sensitive to the needs of third-party development (Kvale & Brinkmann, 2009). In our case, speaking directly to experts in the platform industry and those who have either worked as developers or technical project and development managers is important to validate or extend our theoretical baseline. This approach is therefore suitable given the study is of exploratory nature and the results are often uncertain and dependent on participants themselves along with their previous experiences (Bhattacharjee, 2012).

Interpretive research does however not stray far from subjectivism which means that meaning and words are more important than hard numerical data (Recker, 2013). This implies that the opinions of our chosen subjects, though biased to their own take on what constitutes a successful platform establishment, are still applicable and useful for other similar scenarios. Though it can be argued that quantitative methods are more generalizable in nature, this in our case would not have yielded desired results and would instead limit our findings to a set of pre-determined considerations (Lee & Baskerville, 2003). Identifying how platform companies build developer-centric ecosystems is embodied with complex dependencies. But as Bhattacharjee (2012) argues, interpretive approaches are more suited to explain such scenarios by acknowledging that social reality is often impossible to abstract from its social setting but rather interpreted through sense-making.

3.2 Data Collection

Interviews are the most commonly used data collection method for qualitative studies (Bhattacharjee, 2012). Similarly, we used semi-structured interviews as a guide for data collection as it also complies with the inductive approach and interpretive strategy mentioned above (Myers & Newman, 2007). In accordance with the qualitative method, this allowed us capture rich and in-depth data (Recker, 2013), especially in one of the most important aspects of our study which was to understand what attracts and retains developers in platform ecosystems.

This way we were able to gather data in a less intrusive, but rather a conversational manner that allowed for better clarification and understanding of the subject contexts (Recker, 2013). In this section we implicitly followed Kvale and Brinkmann (2009) seven stages (Figure 3.1) of conducting interview research whereby we first thematized the main concepts from our theoretical review as depicted by the research framework in Table 2.1. Using this, we designed our interview guide (Section 3.2.2) after identifying potential respondents (Section 3.2.1), scheduled and conducted the interviews (Section 3.2.3), transcribed, analysed and reported our findings as seen in section 3.3 and the following chapters.



Figure 3.1: Kvale and Brinkmann's (2009) seven stages of interview research

3.2.1 *Argument for Case / Informant Selection*

Argument for information selection can be related to the act of sampling, which means deciding on which people to interview and why. In our case, we use convenience sampling, expert sampling and snowball sampling (Bhattacharjee, 2012) to identify respondents comprising of digital business model consultants, software development managers or software architects in small, medium to large platform companies who have a holistic view of establishing or running platform-based business models. This way, we are able to have a focused study and to build theory that is largely generalizable to other development contexts and with other types of platform companies. Matthew B Miles, Huberman, and Saldana (2014) justify this approach of having small and purposive samples under a set of boundaries and conceptual framework. Similarly, Seale (1999) highlights the impossibility of exploring all possible scenarios in qualitative research such as to guarantee statistical generalizability but rather achieving the same goal by selecting representative samples that can justify transferability. We do this by acknowledging the diversity in how platform business models can be implemented to suit specific competitive niches. Therefore, we selected 6 interviewees (Rsp1 to Rsp6), 5 of whom represent firms that leverage external innovation in a variety of ways as explained below, and 1 platform expert consultant who understands the current dynamics of developer economics versus the trends in how firms use third parties to extend their business models.

Rsp1 is a mobile web and measurement lead at Google, Ireland. Google is among the incumbents in the successful implementation of platform business models through several of their products led by Android and their backbone technology Google search engine. They continue to launch other technologies such as Google Wallet, Maps and others that ride on the existing ecosystem. Even so, a number of their products have still not gained as much success as the others for example Google Videos and G+ (Fortt, 2006) and thus we aimed to derive some of

the learning points and improvements in such strategies. Having worked for Google for over 4 years at a customer engagement and product implementation capacity, Rsp1 was a useful respondent with knowledge on how platform companies can attract and retain third-party development to propel innovation. He possessed unique insights on how to design and govern for competitive advantage in such an ecosystem and especially from the perspective of a major player in the market.

Rsp2 as an expert in the platform economy is the director and founder of SlashData which was formerly known as VisionMobile, a leading analyst and research company in the developer economy based in London, UK. He has over 18 years' experience in the mobile and technology industry, having worked with brands such as Google, Intel, Mozilla, Amazon, Microsoft and AT&T. He is also an adjunct professor at Lund University where he teaches digital business models and Athens University of Economics and Business where he teaches Entrepreneurship. He was relevant to this study due to his extensive experience and involvement with internet business models and research into developer ecosystems at a consultative basis. His insights were useful in providing an overview of the current motivations of developers in the market and on how companies can tap into this capacity to derive value and competitive advantage.

Rsp3 is a product manager at Direct Pay Online, Kenya with over 6 years' experience in software development and management. As a product manager he is responsible for establishing a product roadmap and designing solutions that best address the customer needs. Direct Pay has a presence of over 30 countries and provides a secure online and mobile payment platform that supports all modes of payments and currencies from major credit companies such as VISA, Mastercard and American Express and all mobile money providers in Africa. Rsp3's insights from working with Direct Pay and other similar companies before were useful for this study due to the fact that the company provides a platform that integrates with other systems in a B2B context through APIs. This way, he was able to give important design considerations when using external innovation to extend a business model which in their case is to integrate as many merchants as possible, in order to reach more customers and generate more network effects.

Rsp4 is a software architect at Jumo World Kenya with over 6 years' experience in system design and development. Jumo is a company that combines technology and data to provide financial services to emerging markets. Through their open API, Jumo platform encourages the creation of predictive data products and models for financial services providers, mobile network operators and other holders of behavioural data sets (Jumo, 2018). Rsp4 has the responsibility to ensure that the domain business model works seamlessly and adapts to changing times. For this reason and the fact they only recently launched the open API, he had good knowledge of the importance of involving external parties to innovate on top of a product and create their own varied use case which would eventually propel Jumo's brand presence.

Rsp5 is a technology solutions manager at Cellulant, Kenya. Cellulant is a company that spans over 11 countries in Africa and offers payment solutions to customers through integrations with banks, merchants, mobile money and network providers. They provide a wallet that enables end users to pay for day to day services via USSD applications. Rsp5 has been a software engineer for over 7 years and now works to design and coordinate solutions for Cellulant that effectively integrate with third-party developers in a B2B perspective. His experience as a developer and as a solutions architect was useful for this study and especially with regard to their recent strategies of launching self-service third-party API that allows for outside innovation and core service consumption.

Rsp6 is the director of software engineering at Mastercard Digital Payments and Labs, Kenya. Mastercard is a technology company in the global payments business connecting consumers, merchants, governments, financial institutions and other businesses in more than 100 countries. Their products enable secure everyday commerce activities such as shopping and other transactions. Rsp6's career spans over 17 years as a software developer, systems consultant and as the head of software engineering at different capacities and industries. His experience in management and development together with his role in Mastercard makes him a knowledgeable participant for this research. Mastercard leverages outside innovation through its MasterCard developers portal where they offer several APIs on security, data services and payments to different categories of third-party developers such as merchants (Mastercard, 2018). As a global market leader, the insights derived from their success strategies in generating immense network effects comes in handy in answering our research question.

Above interviewees represent a great diversity in platform companies in terms of business models and customer reach. Google as an incumbent portrays more flexibility in the relationship with external developers as opposed to master card that has more arm's length partnerships. Jumo as an upcoming middle-sized firm on the other hand has the strategy of providing an open API in order to propel their user reach. These samples therefore demonstrate a representative case selection strategy (Seawright & Gerring, 2008) of the different configurations of a platform ecosystem that enables us to draw insights useful for a wider variety of use cases that involve the use of third-party development to extend a business model. This was in line with case research as one of the interpretive research designs proposed by (Bhattacharjee, 2012) where he describes it as the study of a phenomenon in one or more research sites aimed at deriving and understanding the topic of interest.

3.2.2 *Interview Guide Design*

In order to derive a rich description of the phenomenon of creating a developer centric platform design and governance strategy as perceived by the respondents, we used a predefined interview structure (Appendix 1) guided by the research themes and sub-themes outlined in Table 2.1 above (Recker, 2013). Our questions revolved around the personal considerations of the interviewee's experiences from engaging or researching platform-based business models as guided by the key factors identified in our literature review. Given the time constraints and the potential for these themes to generate a high number of questions, we included open ended questions in each category and allocated available time according to each theme so as to ensure that the discussions were well balanced and not skewed to specific categories. A semi structured interview format allows the interviewees to personally reflect on the subject meanwhile letting the interviewer dig deeper into specific details (Bryman & Bell, 2015).

In order to get a clear picture of the considerations, we used clear and straightforward thematised questions based on 'what' and 'how' while avoiding other speculative and leading 'why' questions. These were also standardized across the different sets of interviews in order to facilitate proper comparative analysis (Kvale & Brinkmann, 2009). With the knowledge that different platform companies have different types of business models, our interview questions were mainly geared towards generic concepts that apply to third-party developer ecosystems as seen in our literature review while avoiding discussions that were specific to a type of product category.

As Recker (2013) proposes, to make the interviewees more comfortable with the discussion, the questions were designed to start with more general ones concerning the respondents understanding and view of the trends in digital innovation and business models coupled with a description of their job roles. These were followed by more specific questions organized to facilitate a conversational approach Kvale and Brinkmann (2009) as per the study themes laddering from developer marketing to governance and control. The last section comprised of a summary question regarding the key and most important considerations as perceived by the respondent in platform design followed by a clarification of any items we may have missed in our themes.

The interview guide therefore was designed with the underlying basis of deriving individual perspectives as influenced by respective organizational contexts (Schultze & Avital, 2011). This was with particular focus on the soft laddering technique (Grunert & Grunert, 1995) in the open-ended questions that allowed the interviewees to have the freedom of expression through relevant examples. The guide therefore provided hierarchical distinctions between key elements of considerations thus allowing the respondents to make sense of their experiences (Reynolds & Gutman, 1988).

3.2.3 Interviews

Following the design of our guide above, we started each interview with the respondents consent to record or not due to varying company policy concerns. However, they were all assured of the confidentiality of their responses (Bhattacharjee, 2012). Since the interviewees were not within proximity, the dominant interview channel preferred was Skype audio and video calls as in Table 3.1 below all in English. The video interviews however exhibited better quality in data collection due to the ability to understand the respondents' non-verbal expressions (Kvale & Brinkmann, 2009). As the interviewees were previously known to us, they felt more comfortable to divulge information as there was already an established level of trust (Kvale, 2006). To save time and ensure better quality of the responses, the interviewees were priorly informed about the purpose of the study and the expected outcomes (Myers & Newman, 2007). The interviews were also planned in advance in order to minimize disturbances and avoid issues like lack of time, which could cause the data to be skewed due to subjects creating opinions under time pressure (Myers & Newman, 2007). Some useful probing techniques as suggested by Bhattacharjee (2012) were also used to elicit more thorough and thoughtful responses such as the silent probe of pausing before going to the next question or overt encouragement using words like "uh-huh" or "okay". At the closing phase the respondents were thanked for their time (Lewis & Ritchie, 2003) and permission sought for further clarification.

Table 3.1: Summary of interview details

Respondent Code	Organization	Position	Interview Date	Interview Type	Duration
Rsp1	Google	Mobile Web & Measurement Lead	25th April 2018	Google Hangouts Video	1 Hour 5 Minutes
Rsp2	SlashData	Director & Founder	28th April 2018	Appear.In Video	42 Minutes
Rsp3	Direct Pay Online	Product Manager	30th April 2018	Skype Audio Call	41 Minutes
Rsp4	Jumo (Just Mobile)	Software Architect	30th April 2018	Skype Audio Call	46 Minutes
Rsp5	Cellulant	Technology Solutions Manager	1st May 2018	Skype Video Call	59 Minutes
Rsp6	Master Card	Director of software engineering	3rd May 2018	Skype Audio Call	53 Minutes

3.3 Data Analysis

It is often too late to begin analysing after the interviews have been conducted (Kvale, 1996). As in Appendix 2 to Appendix 7, all the interviews were therefore continuously transcribed after they were conducted, using OTranscribe online tool into a text document for easy analysis and reference (Bhattacharjee, 2012). The online tool ensured accuracy by allowing for pausing and rewinding of the recording accordingly in order to capture any missed words or unclear sentences. The transcripts were then cross checked for completeness by a different author. To note during the interview process is the use of the memoing concept explained by Recker (2013) to note down important discoveries and interpretations emerging from each interview.

In order to derive patterns in the data and identify key items of significance to answer our research question, analysis was done in two stages using Nvivo qualitative data analysis tool; within-case and cross case data analysis (Bhattacharjee, 2012). In within case analysis, emergent and prominent considerations from the data were manually examined separately per each interview transcription comparative to our research framework. In cross case analysis, similar concepts and patterns were sought between the different interview cases while ignoring some contextual differences in order to generate more inclusive conclusions (Bhattacharjee, 2012). This can be seen in Figure 3.2 below where each transcript was analysed separately, and conclusions drawn based on the number of transcript files and references to a particular sub-theme or concept coded. An example is addressable market, which is seen to have been referred to in all 6 interview files. This process was done iteratively to the point of theoretical saturation where no further patterns could be identified in the available data (Lacity & Johnson, 1994). Drawing inspiration from the grounded theory approach, both stages encapsulated Glasser and

Strauss (1967) data analysis techniques of open coding, axial coding and selective coding as outlined below;

Selective coding: The main themes of our research framework (Table 2.1) formed the basic codes as described in Table 3.2 which were then translated into the base NVivo nodes as seen in Figure 3.2 to guide the mapping of key considerations identified under every section. These in our case marked the key variables expected to be used in categorizing the findings (Recker, 2013).

Table 3.2: Thematic codes

Selective codes / Base nodes	Thematic code description
DM	Developer Motivations
OID	Organizational Identity/ Differentiation
CM	Critical mass strategy
BR	Boundary resources
DR	Design and third-party engagement rules
LC	Lean communication with third-party developers
BC	Balancing control and degree of openness

Open and Axial coding: Each transcript was read through and tentative labels representing meaningful chunks of data created as subnodes relatable to each base node or the core variables as shown in Figure 3.2. Thereafter, causal relationships and connections between the open codes were identified to come up with a final list of considerations under each theme (Glasser & Strauss, 1967). An example from Figure 3.2 below is in the open codes; Fun and Learning, Support, Education and certifications marked under the selective code DM which were linked together to form one consideration named education and support. The same was repeated for all interrelated and overlapping sub-nodes under each node after which they were logically combined to establish final key outstanding factors as reported in the empirical results (Chapter 4)

Software analysis programs like Nvivo does according to Flick (2014) help to efficiently organize, search, sort or process large data volumes based on developed coding schemas. They also, through inbuilt memoing and diagramming functions, allow a researcher to keep track of any emerging explanations and facilitates creation of conceptual models by showcasing relationships (Matthew B. Miles & Huberman, 1984). However, such programs are not able to decipher the meaning or context behind the words and phrases and therefore susceptible to misinterpretation (Flick, 2014). Therefore, since our data quantity was fairly manageable, efficient analysis of the data was possible by going through all the transcript data and code accordingly. We also found guidance in our memo, prior knowledge and our interview guide structure that made it easy to contextualize discussions from the interviews.

Nodes			
Name	Files	References	
DM		0	0
Fun and learning		1	2
Financial incentives		4	6
Organizational efficiency		1	1
Addressable market		6	11
Familiarity with existing software stack		1	1
Locking and switching cost		2	5
Brand equity or market position		4	4
Education and certifications		4	5
Evangelism		2	4
System performanc and usability		3	7
Competitor benchmrking and strategic partnerships		1	1
Ease of integration		1	1
Support		1	2
OID		0	0
One and the same		1	2
diversification of business model		3	6
Strategic alignment		2	2
CM		0	0
Launch like its a product		4	5
Marketing team		3	3
marketing technique		3	7
strategic partnerships		4	7
talent		1	1
BR		4	8
SBR		2	3
direct applications		1	1
Interaction interface		1	1
DR		0	0
Relationship management		4	12
How you use resources like APIs		2	2
Compatibility and versioning		3	4
LC		2	2
BC		5	12
vertical integration		1	1
Outsourcing vs extending bm		3	4

Figure 3.2: Snapshot of coding summary from NVivo

3.4 Research Quality

3.4.1 Reliability & Validity

Reliability implies that the same results are expected to persist even when the same constructs are measured multiple times (Bhattacharjee, 2012). This is similar to the concept of dependability where multiple individuals upon consideration of the same data are expected to reach similar conclusions (Lee & Baskerville, 2003). In this research, this was achieved through analysis of the interview transcripts by two different authors and later agreement and consolidation of the results identified (Recker, 2013). To make sure that the study was consistent and trustworthy, we provided the interviewees with a ‘pitch’ before conducting the interviews. This was done to allow them to get some insight into the study and in turn allow us to gather more precise and relevant data. Accurate transcription of the interview outcomes and the clear documentation of the methods and findings as seen in this report also enhanced dependability of this study by establishing a chain of evidence (Seale, 1999). The contextual diversity in the choice of respondents in terms of the varying platform business models allowed us to find data that would either align or divert from our theoretical review, agreement as well as search for considerations we had not yet discovered (Seawright & Gerring, 2008).

According to Bryman and Bell (2011) validity refers to how well conclusions drawn from gathered data corresponds to the real world. It relates to the correctness, truth and strength of arguments posed with regard to the purpose of the research (Kvale & Brinkmann, 2009). In this study we ensured validity first thorough our research instrument and the usefulness of our theoretical basis (Lee & Baskerville, 2003). Our selected themes and sub themes were consistent with studies and arguments posed by prominent researchers in information systems, particularly on platform business models and innovation networks literature such as Henfridsson, Alstynne, Altman, Tripsas, Yoo, Cusumano and Gawer among others (Kvale & Brinkmann, 2009). The interview questions were consequently crafted from these literature and peer reviewed to ensure the questions were understandable and answerable in accordance to our research question. The key outcomes of this study were similarly illustrated and discussed while grounded on the data and arguments posed in theory (Kvale, 1996).

The choice of interviewees with years of active experience in the industry similarly ensured collection of reliable and accurate data reflecting the current state of events (Lacity & Johnson, 1994). Seale (1999) refers to this as internal validity justified by the credibility and validation of the respondents. In the context of external validity and the generalizability of findings, Lee and Baskerville (2003) argue that qualitative inquiry based on a sample cannot be statistically generalizable but rather only transferable. The selection of diverse respondents from different business models in our case served to limit bias, showcase triangulation through multiple data sources and ensure a good degree of transferability of the conclusions of this study to other types of platform business models (Seale, 1999). A search for disconfirming evidence or negative instances was also considered and notable observations that differed between individual respondent’s discourses identified (Yin, 2003). This was then discussed accordingly, with respect to established theories. An example of this is in the noted levels of importance of the factors identified subject to type of platform business model, which enhanced the distinction between arm’s length and permission-less governance rules. Lastly, the peer reviews done on this paper helped to improve internal validity in the stated accounts (Bhattacharjee, 2012).

3.4.2 Ethics

Empirical studies are often subject to four main areas of concern: harm to participants, lack of informed consent, invasion of privacy, and deception (Diener & Crandall, 1978) summarized by the concept of qualitative ethicism by Kvale and Brinkmann (2009). The conduct of this study consequently followed 4 ethical principles of scientific research outlined by (Bhattacharjee, 2012);

- *Voluntary participation* was upheld by obtaining consent from the interviewees beforehand and letting them select their preferred time and channel of communication. They were also encouraged to use hypothetical examples to illustrate important concepts in the event of sensitivity in incriminating their current organizations.
- The respondents were assured of *confidentiality* in the data collected and recorded to be used only for academic purposes and not disclosed to other parties.
- The purpose of the study was made known to the respondents and details *disclosed* regarding the expected outcomes and usefulness of the results
- Lastly truthfulness and accuracy were upheld in *analysis and reporting* of the findings, especially those that were contrary to our expectations and literature review arguments (Walsham, 2006). Direct quotes were used in some instances to clearly illustrate the empirical results.

The applicable principles of interpretive research suggested by Klein and Myers (1999), such as the principle of contextualization, were adhered to where we were keen to understand the background of each interviewee with regard to how their previous experiences and current organizational contexts impacts their views and opinions. The principle of suspicion was also applied to be aware of any possible biases in the narratives brought about by the specificity and uniqueness of each interviewee's role and type of platform business model (Klein & Myers, 1999).

4 Empirical Results

This chapter showcases the analysis of findings obtained following the methodology applied in chapter 3 above. It describes the results in a synthesized manner highlighting commonalities in what the interviewees said and including some supporting direct quotations as per the transcriptions. The structure of this section follows that of our research framework while including subtopics of the key factors identified under each subheading

4.1 Understanding Developers

Most selected respondents from the different platform companies acknowledge the importance of developers as key digital business model extenders (Rsp2:9; Rsp3:8; Rsp4:10). This confirms as outlined in the theoretical review section the different ways in which developers can be used to create, deliver and capture value. This is also evident in the type of business model exhibited by each platform company where third party developers are either customers in other enterprises, resellers and distributors of the core product or simply as creative innovators deriving new unanticipated use cases (Rsp1:36; Rsp2:31; Rsp3:38; Rsp5:10). To attract, engage and retain such external innovation, platform owners need to understand the needs of their third-party development audience also referred to as developer motivations (Rsp2:19; Rsp5:14).

4.1.1 Developer Motivations in Platform Adoption

These will advise platform owner strategies for targeted marketing and on how to lock in and raise switching costs to prevent developers from moving to other similar platforms (Rsp4:14). Rsp2 (13) explains that this has enabled technology brands to be ranked based on adoption, satisfaction and engagement rates. Such motivations are outlined below;

Addressable market:

Having a substantial audience to launch innovations remains top-priority as all of the respondents agreed that addressable market or customer base to be the most important factor of developer motivation (Rsp1:8, Rsp2:13, Rsp4:14). Being able to attract developers relies to a high extent on knowing how big the current market is and the subsequent growth potential regarding user reach (Rsp1:22). Recent events have shown that prominent and hyped technologies that fails to provide a significant addressable market will fail to really attract developers. Rsp2 (13) gives the example of VR technology, where despite extreme hype complementary development has never been able to really take off. Rsp2 (13) claims the reason being that core devices never targeted markets of hundreds of millions of users, making the addressable market too small to spark third party development. Certain products also differ in customer reach on a regional perspective as Rsp6 (9) gives the example of how different markets have different dominant players as in the case of WeChat in China. This therefore will influence the choice of platform by respective developers as they aim to reach more customers.

“there is another aspect that is probably the most important for anyone who is not a hobbyist, and that is the addressable market, in other words if a technology doesn't allow you to reach users, then it immediately loses its appeal”

-Rsp2 (13)

Brand equity:

A sizable addressable market does however imply more than a large base of potential customers. Vast user-reach means a wider adoption of technologies which in turn increases the value of the personal brand of the developer (Rsp1:8; Rsp2:15). Rsp1 (8) and Rsp4 (10) state that developers want to be associated with a brand that has a strong recognition and adoption of technologies. Those with experience and expertise in technologies supported by brands that are considered thought-leaders in their market space are more attractive to a wider audience than specialists of less-known brands (Rsp1:8, Rsp4:10).

“There is like a brand equity aspect to it that motivates a developer that wants to be associated with that brand. To market themselves as consultants that offer this type of expertise.”

-Rsp1 (8)

Compatibility, interoperability and strategic partnerships:

This has to do with cross platform compatibility in terms of architecture and operating system considerations. Rsp3 (14) and Rsp5 (34) extend this especially from a business perspective, stating the importance of constantly innovating to keep up with latest technologies and securing partnerships with other leading platforms. An example is incorporating into a product payment solution from leading financial services providers such as AMEX, VISA or Mastercard that already have established market leadership (Rsp3:23). This appeals to developers in a holistic manner both from a technical compatibility perspective and a brand equity perspective. Another way is in using mega-SDKs which contains diverse types of technologies that work well together that the developer will end up using technologies from only one vendor due to compatibility reasons (Rsp2:21). Compatibility and simplicity is therefore an important used selling point and lock in mechanism for developers (Rsp2:21).

“...what we try to do is to always innovate around the solution that we have and as well, getting to know what your competitor has on the ground, knowing what they are working on and trends within the market and then also we have the strategic partnership...”

-Rsp3 (14)

“...mobile companies are building mega-SDKs. As in SDKs that combine so much functionality in it, so that you end up using more and more of this from the same company and in the end of the day you end up using only Microsoft or amazon or Google technologies.”

-Rsp2 (21)

Direct monetary incentives:

While monetization is of the highest importance especially in B2B contexts, direct monetary incentives are not necessarily used as the major means of motivation for third party developers by the platform leaders themselves (Rsp1:8; Rsp2:11; Rsp4:8). Usually, the transactional phase where the developer gets paid is done by the organization aiming to use the third-party platform expertise and outside the platform hosts direct sphere of influence (Rsp1:8). Neither Google nor MasterCard offer direct monetary incentives on their platforms, instead they support developers in mastering their APIs which makes them more likely to get hired organizations that are ready to pay for external innovation (Rsp1:8). However, the concept of pricing is unavoidable

especially with the economics of a willing buyer and willing seller where both parties are motivated by cost and benefit as seen in revenue share models (Rsp6:5; Rsp3:38; Rsp5:20). Rsp3 (38) gives the example of a tap model they use to lock in merchants who integrate into their platform whereby they give incentives to merchants in such a way that the more transactions they register per month, the less service charges incurred.

“we do not give monetary incentives, the incentive we give in most cases is certification”

-Rsp1 (8)

“we use what you call a tap model. we have a standard charge for merchants who have low volumes, but we give them an incentive if they grow their transaction”

-Rsp3 (38)

Developer education and support:

Support and education are two different yet closely related aspects of developer motivation. All respondents agreed that these two factors should be prioritized and should be continuously invested in as the platform grows (Rsp1:38). Showcasing platform properties through education is an important form of developer marketing so that developers can learn, try, adopt or even contribute to the product (Rsp2:19). As the commercial success of those offering their innovations and technology, largely determines the success of the entire platform, platform hosts invest heavily in providing educational tools for its producers (Rsp1:40). There are numerous ways to do this and some of those that have been listed by respondents are hackathons, training workshops, social media channels such as blogs, hands-on labs, online tutorials, conferences and other events (Rsp2:19). Such initiatives attract developers onto the subject ecosystem through peer motivation that encourages to learn and use a product. This however does not rule out the personal initiative of those willing to develop against an API for a company or product that they wish to work with in the future, just for the sake of gaining experience in context of the job market and career advancement (Rsp4:8; Rsp3:6). Some companies also incentivize by giving product certifications that confirm developer expertise on certain products thus giving them more credibility and recognition (Rsp1:8).

“As a young, upcoming developer you strive to reach a certain point and look for a mentor who you feel is much better than you from a technical perspective...”

-Rsp3 (6)

“...the incentive we give in most cases is certification. you are able to have a certificate that means that you can offers this kind of expertise in for an example android to companies that are seeking that type of skill.”

-Rsp1 (8)

Support on the other hand ranges from having on-site consultants to providing documentation, answers in public forums, design playbooks from free user research and more formalized training to third parties (Rsp1:38; Rsp2:19; Rsp5:50). Once a developer has gained expertise of a specific platform or product, the platform owner may offer support by facilitating the individual to train other developers in that specific niche area or by empowering them through business recommendations or sponsorships to attend product launches and other capacity building events (Rsp1:40). Another approach to driving developer engagement, which is more distinct from

traditional marketing efforts, is evangelism (Rsp2:19). Evangelists are sales-people in disguise that aim to drive platform transactions by supporting producers in their business models (Rsp2:19). Evangelism is done through a breadth or a depth approach (Rsp1:12). In the breadth-approach frontline developers are supported by developer advocates who help them overcome technical issues and optimize their use of platform technologies. In the depth-approach, c-suite managers= are approached by the platform host in a more consultative manner (Rsp1:12). The platform consultants aim to advice the managers on how to strategize the business model around the platform technologies and is thus driving incremental revenue through the platform's own business model (Rsp1:12). Support could also be in terms of how long it takes for a developer to get a response after logging a system support ticket, or the availability of a test environment where integrators can experiment and test their ideas before transferring to live environments (Rsp3:34; Rsp5:50).

“...if you are advertising Google but you have a very high bounce rate, it means you as a business do not benefit because you are not getting conversion for sales and us as Google will not benefit as well so we provide this as a free ad on service for our large advertisers to help them develop best sites with best principle...”

-Rsp1 (38)

“What we normally make sure is that accessibility of the system that the users even if you provide them with the documentation of the system then you should be able to provide them with a staging environment where the integrators will be able to connect to the system and confirm that they are able to consume the services”

-Rsp5 (50)

Language agnosticism:

One aspect of simplicity that has become more prominent in recent days is allowing developers to use their stack of preference for development. Rsp2 (11) states that we have a tendency as people to create emotional connections to things that we build and in the development community this is known as technology tribes. This implies that developers tend to stick to the programming language they are used to. To attract a wide range of different developers, successful platforms therefore need to take an agnostic approach to programming languages (Rsp2:25). A bigger selection of available languages and stack allows for a bigger target group of developers. This means that language locking strategies as previously used by Nokia and Microsoft in C# is no longer appealing to developers who value flexibility (Rsp2:25).

“...you develop emotional connections to things that you have built yourself, so you spend a lot of time working in...”

-Rsp2 (11)

“modern platforms are very language agnostic, (...) supporting all languages and the developers using it were coming from all language backgrounds.”

-Rsp2 (25)

Usability and ease of integration:

Developers, like any other users, value a great user experience in terms of how easy it is to interact with or use the solution provided in terms of procedural steps and even aesthetic appeal (Rsp5:34). This has to do with the complexity and relevance of the procedures of integration (Rsp6:59). It is of consideration for the third-party developer to look at how the provided interface with the core product serves the intended purpose efficiently through aspects such as self-service reports that depict transparency of interaction activities (Rsp5: 24). Noted is also the technical complexity involved for example, some legacy systems have a lot of instructions and commands on how to integrate as opposed to more current technology implementations that ensure seamless integrations that adjust accordingly (Rsp5:34).

“...the ease of integration, I think there are some systems especially the legacy systems it becomes very hard to integrate with those systems because they have so many instructions and commands on how to integrate with the system compared to maybe a restful api that adjusts, its easier to integrate with that system.”

-Rsp5 (34): on what would make developers choose one platform over another

System Performance:

Other than the ease of use and integration, the performance of the system is a factor that developers consider (Rsp3:36). This has to do with speed or turnaround time, reliability, load handling, monitoring security and accuracy of results and transaction processes (Rsp5:34). Having a history of security breaches impacts on the overall attraction of the product to any potential integrators especially where financial transactions or sensitive data is involved (Rsp6:31). Platform owners therefore seek to ensure that all services are up and running always effectively and efficiently so that developers do not turn to a competitor with better performing systems (Rsp3:14, Rsp5:34).

“...a client (would) go to a competitor simply because your service is not working as it's supposed to be so essentially your competitor is having an edge over you.”

-Rsp3 (14)

4.2 Platform Design Strategies

Besides developer preferences, organizations need to make strategic and technological considerations when entering the design phase of the platform. Respondent 1-6 have all discussed the delicate state of the launch process and actions aimed to reach critical mass. Building on the insights of developer motivations, respondents as in this section describe various ways of strategizing distribution of core platform products and the subsequent impact on the organization's identity.

4.2.1 Boundary Resources

APIs and SDKs:

APIs and SDKs constitute the most foundational boundary resources that are used in every business model open to third-party development (Rsp1:34). Due to their vast diffusion over most platforms, the resources have become a point of diversification (Rsp4:30). Rsp4 (30)

claims that to create the best and most attractive platforms and ecosystems, organizations must treat their API as products and developers as the potential customers. However, the available boundary resources determine how and what is going to be developed on your platform. Rsp1 (38), Rsp4 (36) and Rsp6 (31) all agree that APIs and SDKs should be easy for third parties to use in terms of seamless design and overall usability.

“Yeah i think at some point i said you need to look at api as a product and if you do that then all these considerations become very relevant in terms of how you package which means that it has to have a nice UI, the developers are able to interact with it, you have a good support mechanism ...”

-Rsp4(36)

Social boundary resources:

Boundary resources do however span beyond APIs and SDKs. Rsp5 (66) notes that in some instances rigid contractual agreements are used to manage third parties and formalize what happens in the system. Contracts or terms and conditions and non-disclosure agreements also have the benefit of clarifying what the developers can expect of the system and what the system allows the developers to do with it (Rsp5:52). When opening to third parties, the sheer size of the developers included in your network can however make it impossible to create customized contracts for each and every one (Rsp5:66). This is especially true for permission less innovation (Rsp6:37). Organizations therefore create different kinds of contractual agreements depending on the profile of the developer (Rsp5:66). Platform organizations with a lower degree of openness have more arm's length relationships hence are more focused on creating customized contracts for larger key accounts, who typically have special requirements on how they interact with the system (Rsp5:66). Other social boundary resources include developer advocates, product consultants, account managers or even student ambassadors who act as representatives or evangelists of the core products value propositions (Rsp1:12). Such resources have the responsibility to for instance organize hackathons, conferences, incubator programs, trainings, maintain blogs and social media forums among other support and marketing tasks (Rsp2:19). They thus play a big part in ensuring developers are attracted, engaged and retained in the platform.

“Contract binds [developers] to make sure that we agree on how they use the service and those include our expectations and what they can expect of the product.”

-Rsp5 (52)

“...data contracts we usually have with key merchants and these big companies. then we have like terms and conditions for smaller actors and individual developers. when you open the system to the third-party developers, we can have an endless amount of developers and it's hard to create a customized contract with each one of them.”

-Rsp5 (66)

Direct applications:

An interesting trend as reported by Rsp1 (34), Rsp3 (32) and Rsp5 (10) is the strategy to simplify applications such as they offer plug and play capabilities to a third party who may not be

have coding experience. These include for instance merchant facing programs that offer a standardized method of utilizing core platform capabilities (Rsp3:32). They offer a configurable application targeted to certain profiles of third party developers (Rsp3:32, Rsp5:10).

“...there is the checkout platform which is what we give to third party companies or individuals that want to integrate to our system, and third party they will be able to charge the customer and fulfil the request of the customer. what we normally do is we give you the platform and you give us your customers then we do a share basically which could be 50/50 or 60/40 kind of sharing in terms of the commission...”

-Rsp5 (10)

4.2.2 Critical Mass

Launch and talent acquisition:

The launch phase is crucial for the future success of a platform. To gain both an adequate number of producers and consumers, the platform owner usually needs to be able to sustain financial losses at the beginning while maintaining maximum product impact (Rsp6:25; Rsp3:28). Rsp3 (28) underlines that you must move quickly in digital business to either get first mover advantage or be able to disrupt incumbents in an existing market. Rsp6 (23) emphasizes this by indicating that having a unique product that sells itself by solving someone else's pain point goes a long way in facilitating attainment of critical mass. Facebook is noted to have perfected the art of orchestrating the launch of new open source projects to have maximum impact on day one much like how other incumbents like Google and Apple do in their yearly launch events (Rsp2:41). In fact, Rsp2 (41), Rsp4 (36) and Rsp6 (23) all agree that releasing an API should be approached just as one would in launching a product. First mover advantage and innovative disruption is not achievable without the right talent at the right time (Rsp3:40). Not having this strategically planned initially would otherwise have negative cost implications in terms of the hiring and learning process given the specificity of technological talent (Rsp3:40).

“So, Facebook has perfected how they launch new open source projects and gain maximum impact. not maximum but very strong impact from day one because at their conference they were showing charts whereby the overall impact to developers of a new open source project is very much dependent on the impact on day one, so you have to get the launch on day one exactly right in order to have a lasting effect...”

-Rsp2 (41)

“...a starting point for certain businesses where they have to look for a certain kind of talent, which becomes very expensive so if that kind of talent lacks, they may have to hire a lot of people and then start doing the learning process.”

-Rsp3 (28)

Strategic partnerships and product bundling:

Reaching critical mass implies quickly being able to offer a wide range of services to both producers and complementors in the ecosystem (Rsp3:22). Unless being a transnational incumbent as the likes of the GAFA-companies, platform leaders need to establish partnerships with other platforms and ecosystems to provide the array of services that retains developers (Rsp3:14). This especially comes into play when aiming to pursue the business model for the

first time or aiming to penetrate a new market. Rsp1 (22) exemplifies that when Google moved into Africa, they made sure to initially identify market leaders, and integrate Google services into their platforms. This is confirmed by Rsp5 (16) who's company services have been leveraged by such external incumbents aiming to drive their products locally. The high adoption rate of android was also propelled by strategic partnerships where Google initially offered it to handset manufacturers for free to drive a high customer base and in turn attract diversity in third party developer applications (Rsp1:22; Rsp4:28). Company acquisitions and mergers also come in as a recent strategy to gain critical mass on a product without the logistics of developing the solution from scratch (Rsp1:32).

"...that's just some example how we go about using existing ecosystems to grow user base of new products. But it's different ways in how we go about driving user adoption. If we go to Africa for an example, we don't have the market share like we do in Europe or in America. We have existing local companies that have existing companies that have better market share than we do and, in such cases, how can we support these partners in a way that aligns with our business model?"

-Rsp1 (22)

Developer Marketing:

Simply developing a product is not enough. The third-party developer community needs to be aware of its existence to make the choice of whether to adopt it or not (Rsp1:12). This aspect has earlier been referred to as developer marketing but in this context a focus is made on the initial roll out (Rsp1:12; Rsp6:59). These could involve early access programs, reward programs and developer segmentation depending on the type of third party developers a platform aims to target, in line with their business model (Rsp6:59). Most respondents also agreed that these initiatives do require a separate dedicated team that specializes in product marketing through channels such as social media, mainstream media, sales representatives, email campaigns, billboards, and other advertising channels (Rsp1:12; Rsp5:46; Rsp3:8). Pricing models also do have appealing elements to developers for instance in having an initial commission share that greatly benefits whoever integrates to the platform hence drive adoption through financial incentive (Rsp5:46,20). Other methods noted include growth hacking techniques used by technology companies and product bundling where different related services are integrated into one solution (Rsp3:24, Rsp1:22).

...according to what the strategy that has been laid down is that in the end we will be able to push this through media and what we are going to do is that the commission share is going to be either 40/60 or 30/70 and this is the advantage of the guy connecting to our system, meaning that they connect to our system they make more money."

-Rsp5 (20)

4.2.3 Organizational Identity and Differentiation

Diversification and the core product

The evolution of technology is inevitable, and every company continuously innovates around their core product to keep up with current trends such as in big data and AR/VR (Rsp1:28; Rsp4:20; Rsp3:14). Even with the motivation to stay focused on one's core business, tapping into different kinds of business models remains unavoidable to ensure increased profits, market

presence and simply what is referred to as sudo-ego (Rsp1:28; Rsp4:20). An example is given of an incumbent that decides to monetize its extra infrastructure by offering cloud services to other businesses in order to drive incremental revenue (Rsp1:28). Such success is often enhanced by the already established market thought leadership in the original core product that serves to drive adoption of consequent products and services (Rsp1:30; Rsp4:20). Such diversification however often could result in chaotic management structures that could threaten a company's core revenue stream through lack of focus (Rsp6:17; Rsp1:30). A common solution seen is often the creation of a conglomerate to act as the umbrella or governing body for all the different focus areas with each area having its own management structure (Rsp1:30). This brings to light an important consideration of the process of when a company hits threshold in its core innovation, it looks into other enhanced prospects in what is called adjustment innovation and finally into disruptive innovation in pursuit of further differentiation (Rsp1:30). This therefore has to do especially with the size of the company since the more it scales the more it moves out of its core business and seek other revenue generating initiatives (Rsp6:19).

There have been instances where the transition into a platform business model has impacted the overall identity of the firm to a much greater extent. Rsp2 (39) gives the example of Android and iOS that can now not be imagined as being "platform-less" but instead are one and the same thing. Others like Stripe and Twilio cannot exist without the platform business model which takes up a big part of their brand identity such that the platform is the product (Rsp2:39).

"...we know that our core is advertising but we don't want to be left out once something is happening around the world. Ideally, we tap into any kind of business model...once the core innovation hits threshold you have to look into other stuff and once you got an adjustment innovation right you have to look into disruptive..."

-Rsp1 (30)

Strategic ecosystem alignment:

Retaining the developers primarily depends on establishing a large base of end users that the developers can market their technologies to (Rsp4:14). End users, in turn, want the service that the ecosystem provides to be as simple and efficient as possible meanwhile having all services needed available in the same space (Rsp3:14). This requires platform leaders to sometimes think outside the box, and make sure that every aspect of the journey from early development to end-user distribution is included in their platform. If the platform leader does not possess the often-vast capital needed to provide all the required services, integration with other ecosystems is a must to remain competitive (Rsp3:22). The alignment of both internal and external stakeholders through market research and advocacy is also essential in ensuring the new product acceptance (Rsp1:48).

"...getting to know what your competitor has on the ground, knowing what they are working on and trends within the market and then also we have the strategic partnerships with AMEX, visa and Mastercard so any new innovation they come up with you get to be the first to use..."

-Rsp3 (14)

4.3 Governance and Control

4.3.1 Platform Rules

Consistency:

The dynamics of an ecosystem imply that any changes made on the core product are bound to affect all actors (Rsp5:62; Rsp6:43). Companies however still need to innovate but are especially motivated by end user satisfaction, hence the need to always do user research to see what changes will have a positive impact on the market (Rsp1:24). It is clear from the respondents that the platform owners always have an upper hand on how to evolve the product as motivated by competitive advantage (Rsp1:24; Rsp2:53). Rsp2 (53) gives an example of Symbian which made a critical change in the codebase requiring developers to re-write their apps. This turned out to be one of their major recipes for their downfall in the market. For this reason, several respondents (Rsp2:55; Rsp4:46; Rsp6:43) are seen to bring up the concept of backward and forward compatibility and versioning of the APIs to maintain trust with the developers that what they build on a platform will be durable for the platforms lifetime. Noted however especially in highly contractually governed relationships and transactional systems is the importance of communicating any internal system changes and maintenance activities to all partners (Rsp5:62).

“...the apis you take backward compatibility or in some cases there is forward compatibility. so backward compatibility means that the new apis will be a superset of older ones.”

-Rsp2 (53)

“...whatever changes are made to not impact previous versions, so its like you post the new version and that stands on its own and the old version is still working with full access to its functionality so unless there is a security issue...”

-Rsp6 (45)

Relationship management and resource use:

The use of APIs as a boundary resource in the interaction with third party applications, requires compliance to some set standards such as design that takes into consideration the parameters that will be exchanged such as personal identifiable information (Rsp1:46). The platform therefore needs to vet every partner that comes on board with or every application before incorporating into the mainstream product (Rsp1:46; Rsp3:36; Rsp6:7). The vetting process as is made clear is greatly influenced by the type of business model, scaling strategies and type of partners of third party developers targeted. Financial services firms tend to emphasize more on security measures, and in some cases offer API portals in form of sandboxes that do not warrant immediate live integration but rather experimental avenues for potential partners or business model extenders (Rsp6:7; Rsp3:36). Additionally, the context in which the third party is coming in is also considered to enforce partner level security on top of the product level security measures (Rsp6:33; Rsp3:36). In other instances, such as Amazon however, the API is publicly available to be used in whatever way one pleases and anyone is welcome on board with minimal governance save for terms and conditions (Rsp2:49). A common denominator for most respondents however as mentioned under boundary resources is the use of appropriate contractual agreements and terms and conditions to manage engagement with the third parties (Rsp5:66).

“What happens is that there are several layers, first the product that your aiming to develop is secure, the second is the context you're coming in, through which partner and that means that you are also limited further in what you are explicitly given access to. there's a product level security and then there's a partner level security...”

-Rsp6 (33)

4.3.2 *Balancing Creative Freedom and Control*

Degree of openness:

Findings indicate that there is no one right way to do this. Industry expert Rsp2 (47) in fact states that *“it is an art rather than a science”* and that *“every company choses where to draw the line”*. He states the importance of understanding your own business and what to expose, to which audiences and with which priority. Everything invented cannot always be actualized and more openness equals more risk but more positive innovation at the same time (Rsp2:47; Rsp6:41). Referring to a published report on open governance index, all companies whether open source or proprietary exhibit a certain degree of openness (Rsp2:29). An example given is of Apple which initially only wanted developers to build native apps through their native apis and not web apps which are easily portable to other devices to maintain control (Rsp2:47). Rsp5 (30) however portrays the dynamism of such decisions by a company citing a local Kenyan mobile payment services provider called Safaricom that initially had closed their services and APIs to specific integrators only to later open it up to anyone who wishes to incorporate mobile payment services onto their application. Another interesting finding is in the use of sandbox or staging environments tied to specific products that offer access to development ideas without direct flexibility to actualize it in the production environment (Rsp5:50; Rsp6:49). The degree of openness therefore changes at the product level depending on type of business model. Most respondents however tended to prefer more open platform approaches despite the stated challenges posed by organizational identity, processes and type of business model (Rsp3:40; Rsp4:16; Rsp6:55; Rsp4:24).

“... there are two levels, there's access to the ideas for development which is mostly open, where you have access to a sandbox, but the challenge is how to actualize this in the real-life environment? that's where it becomes difficult...”

-Rsp6 (49)

“...you can still have open source (systems) and still make business for yourself or the company...”

-Rsp5 (30)

Control through partnerships and policy compliance:

This is exhibited by how Google controls android by establishing strategic ecosystem partnerships with low cost handset makers, telecom operators, accessory makers and app developers through the app store (Rsp2:51; Rsp4:26). An element of vertical integration is also evident in ensuring quality control as seen in Google's development of the pixel phones to enhance the standards of distributed software as well (Rsp1:36). It is also stated that the process of developing a product in a company involves working with cross-departmental teams such as legal, marketing, finance etc. to get input on how best to shape the product. Legal counsel particularly stands out as a key advisor on what dimensions of the product to expose in the API in order to

avoid infringement of company's data(Rsp1:48). The aspect of legal representation is therefore unavoidable in this context as implementation of technologies requires compliance with set policies and privacy laws of an operating region for instance GDPR in EU (Rsp1:48).

"... let me give you an example. android was not the first open source mobile OS there were others before it. but what google did, they said to the manufactures we are going to give you this thing for free, no strings attached. have it, install it in your devices and that essentially sparked developers to develop for android because suddenly every device that was coming to the market had android installed"

-Rsp4 (26)

Outsourcing versus digital business model extension:

Noted is that the concept of outsourcing to third party developers is differentiated from extending a digital business model using third party development. The former requires caution not to externalize one's core business model to the extent that the third-party service provider can easily replicate it and resell the same services (Rsp1:56). The latter as discussed in this paper however is a more viable approach to scale a business as it leverages the ecosystem by encouraging more producers to participate and co create on the platform while safeguarding the key competitive product advantage (Rsp1:56; Rsp4:26). Further, according to Rsp4 (24), when a firm opens up more to outside innovation, there are advantages in the public scrutiny and peer critiquing which can be used to improve the core product and avoid tunnel vision. However, others may discover potential vulnerabilities and use that against the company (Rsp4:16). A factor that is noted to be beneficial in outsourcing is the aspect of managing scale. As a company grows, so does the workforce brought about by the increased ecosystem load which could necessitate the use of consultancy companies such as McKinsey to advice on management strategies e.g. evolution of contractual agreements to suit the new scale economics (Rsp1:28).

"...you can read about companies that have used third party developers (outsourcing) and ended up even losing their own business model because this developers can create the exact same things.."

-Rsp1 (56)

4.3.3 Communication Structures and Leadership

The aspect of lean communication for knowledge sharing between the platform actors i.e. third-party developers as a way to engage them and facilitate collaboration was an important aspect acknowledged by all six of the respondents. The strategies that stood out include;

Developer ecosystem networks:

This is a structured approach that fosters voluntary participation of interested developers based on regional location. Facilitated by a leader who could be referred to as an evangelist, the actors in this ecosystem get support from the organization to enable sharing of knowledge and problem solving in context of the core product (Rsp1:52,54). This can be done through events and other social meetups purposed for the advancement of the host platform.

"This is a community of developers interested in developing for google products and embrace google technology purely on a voluntary basis, if you want to learn more about using google apis"

and such google would empower you with tools and events like hackathons...”

-Rsp1 (54)

Technology channels:

This includes the use of communication channels such as real time community forums and mailing lists where members can ask questions, interact with one another and the platform host for support or development (Rsp4:40).

“Most open api specifications or implementations have what they call mailing lists, you have community forums where people can actually ask questions, and get help from the rest of the community, so there exists such kind of communication channel to use.”

-Rsp4 (40)

5 Discussion

This chapter aims to synthesize and discuss how empirical findings and literature relate to one another. Since third party development is best analysed as a process, where considerations are built on and added along different phases of the platform's establishment, the discussion follows the same structure as the theoretical background. The findings are compiled into an enriched version of the research framework, where new additions to previous studies have been highlighted. Additionally, the five factors of DOI help explain how each consideration affects platform diffusion rate. Considerations have hence been discussed in context to corresponding DOI factors. It is evident that all Rogers (2010) factors are considered by respondents, despite their varying business models. Thus, it can be presumed that hosts expect developers to approach platforms as prospective adopters of new technology. A comprehensive list of key factors given all stated considerations and discussions is outlined in Table 5.1 at the end of this chapter.

5.1 Developer Motivations for Platform Adoption

Developers do undoubtedly represent the key actors of platforms. In accordance with Tura et al. (2017), respondents acknowledge that their needs and preferences act as the base on which future design and governance considerations will build upon (Rsp1:36; Rsp2:31; Rsp3:38; Rsp5:10). Understanding developer motivations thus serves as the initial phase of platform establishment.

Five out of six respondents claim that the highest motivation for developers is the addressable market covered by the core product (Rsp1, Rsp2, Rsp3, Rsp4, Rsp5). Addressable market is thus the factor that seemingly gives a platform its strongest relative advantage to other platforms and is hence the most prominent driver of platform diffusion (Rogers, 2010). This appears to supersede the aspect of financial compensation as suggested by the studied literature (Hsieh & Hsieh, 2013; Vannieuwenborg et al., 2012). The importance of the addressable market seemingly goes beyond the financial benefits of a large customer base, or the curiosity of building on the latest technology. Rsp4 (14) in his statement “...it all depends on the potential customers that for example third party developers will see in your platform, so as long as you ensure your platform has enough number of partners and customers, naturally developers will go towards places where they have bigger impact...” justifies this. Rsp2 (13) also claims that the value stems from innovating a technology that can be as widely adopted as possible and, in the future, maybe even centre its own ecosystem. Rsp2's example of VR technology portrays this phenomenon since being one of the latest and hottest technologies that has been the subject of extreme hype, there is still a clear lack of interest from third parties to develop on current devices. This draws similarities to the discussion by Dellermann et al. (2017) who underline that market conditions like addressable user base is crucial for third-parties to join a platform.

However, despite a product having sufficient addressable market to attract third-party development, it is impossible to ignore the element of brand equity, whereby developers tend to prefer being associated with technology brands that exhibit thought leadership as in this statement by Rsp4 (10), “...most developers are going to develop for google api because they are well known...”. Budac and Baltador (2013) define brand equity as the cumulative value of a brand

that makes it identifiable or easily recognizable to the public. The example given is on the challenge upcoming companies may have in promoting their APIs to developers and the need to establish unique value proposition to supersede the general preference in current market incumbents (Rsp1:8; Rsp4:10). This indicates that when faced with a decision where two companies both offer the same product and with the same amount of addressable market, a developer would most likely choose the company with a stronger brand equity (Triche, Cao, & Thompson, 2013).

Interestingly, literature has rather emphasized the role of monetary incentives in third-party development (Boudreau & Lakhani, 2009; Hsieh & Hsieh, 2013; Shah, 2006). For an example, Hsieh and Hsieh's (2013) argumentation revolves around the correlation between developer commitment and monetary rewards. It insinuates that one of the most important considerations for those aiming to build an ecosystem with third-party developers, is how to get them paid. Naturally, there will be differences between permission less and arm's length relationships. In business to business context, getting paid is undoubtedly a very important consideration (Lakhani & Von Hippel, 2003). This stems from relationships almost always being contractually bound and hosts strategize on how to use financial incentives to lock in partners beyond contracts. Rsp3 (38) gives a good example in describing how customers face lower charges if they do more transactions on the platform.

However, when it comes to individual and permission less developers, findings show that some of the most successful development platforms refrain from giving developers direct monetary incentives as stated by Rsp1 (8) "*...we don't give monetary incentives, the incentive we give in most cases is certification...that you can offer this kind of expertise in for an example android to companies that are seeking that type of skill*". Here we see that instead of directly paying developers for their efforts, the main value offering comes from supporting developers in building their expertise and in turn their own personal brand. This highlights another aspect where the empirical results take a slightly different route from the literature as seen in Accenture (2018) report. As a strong and widely known brand usually implies a large addressable market, it constitutes a selling point far greater than most pricing models that are directly connected to the innovation itself (Rsp1:8; Rsp5:20; Popp & Meyer, 2010)

Since developers prioritize building their expertise, they have high expectations of the platform's training opportunities. Hosts consider what resources to offer that will increase testability meanwhile decreasing complexity of the platform (Rogers, 2010). The importance of educational tools, seminars, documentation and formalized training was stated by all respondents and aligns well with literature (Free, 2018; Midha & Bhattacharjee, 2012; Vannieuwenborg et al., 2012). Rsp2 (11) makes an interesting point in his reflections of developer's often emotional connections with the programming language they started off with. This implies that platforms that support a high number of different kinds of stacks and provide the support that caters to various programming preferences are more attractive to developers. Rsp5 (36) supports this as he states that in order to retain developers in the ecosystem, they must have access to all services, tools and technologies they need to take their innovation from early development to end-user distribution.

Support in form of customized on-site training stands out as generally highly appreciated developer motivation (Rsp1:38; Rsp6:1). Providing direct support enhances the developer experience, meanwhile opening up for "soft sells" where the host can showcase additional technologies in the ecosystem (Rsp2:19). This is an efficient strategy to increase observability of what value the platform can offer if integrated in the proper way (Rogers, 2010). Additionally, Rsp2

(21) described this as an efficient way for hosts to lock in developers in ‘mega-SDKs’ that prevent them to seek out rivalling ecosystem due to compatibility reasons. Ensuring that all services are provided often requires hosts to establish strategic partnerships and enable technology sharing to ensure compatibility between platforms and ecosystems (Rsp2:21; Rsp5:34).

Lastly, several respondents have highlighted the impact of technical aspects such as ease of integration and system performance on developer adoption (Rsp3:36; Rsp5:34). Being in the financial services industry, Rsp6 (41) in his statement “...when it comes dealing with people's money then the bar changes coz it means you have to protect the reputational risk that would come from people saying , beware of their products , there are few fraudsters that deal with that platform...” shows that a history of security breaches or poorly running systems will have a significant impact on third parties willingness to integrate with the platform. Findings align with the studies by Free (2018) and Accenture (2018) which show that developers simply do not wish to work in slow or unreliable environments. Rsp3 (36) and Rsp5 (34) emphasize the importance of minimal downtime, a consideration that is amplified when the platform extends its reach to developers in different time zones or with different technical infrastructure. This therefore has to do with what Hasselbring and Reussner (2006) refer to as the trustworthiness of software systems.

5.2 Platform Design

Following the phase of understanding developers, platform owners need to design the product in accordance with developer preferences to appeal to them and drive adoption. This stage includes the design of boundary resources which as per the findings above is often linked to particular product, launch strategies to attain critical mass and the implications on organizational identity and differentiation.

5.2.1 Boundary Resources

To start with, APIs and SDKs remain the foundational boundary resources of third-party development and it is today nearly impossible to find a platform not containing both. This can be related to Rogers (2010) factor of compatibility that impacts adoption of innovation by seeking to streamline the interface between the core product and external innovators. Similarly, it positively impacts the degree of observability as capabilities of the system can be showcased. Due to high diffusion, respondents underscore that APIs and SDKs now need to be treated as products in regard to diversification and marketing (Rsp4:36). Developers are attracted to APIs and SDKs the same way customers are attracted to products and how to diversify and communicate thus becomes an important consideration for platform hosts (Rsp3:14). Thus, findings indicate that organizations have begun to realize the issues raised by Mohagheghzadeh and Svahn (2016) and are trying to minimize discrepancies between available and perceived boundary resources on their platforms. This includes a well-designed UI to meet developer demands of simplicity and compatibility, supporting the notions by (Nylén & Holmström, 2015).

Nonetheless, despite Benzell et al. (2017) justification of the impact of APIs on a firm's performance, findings support Ghazawneh and Henfridsson (2013) claims of traditional APIs and SDKs not being enough to attract and retain developers (Rsp1:21; Rsp4:30). Social boundary resources are needed to maintain developer relationships outside of the technological sphere

(Ghazawneh & Henfridsson, 2013; Youngjin et al., 2010). Rsp2 (19) exemplifies hackathons, conferences and social media as social boundary resources allows for the essential ‘tuning’ of platform resources, as highlighted by Eaton et al (2015). Additionally, satisfying developer needs for personalized training through different types of evangelism is an interesting and seemingly highly effective relationship management strategy. Rsp1 (12) in his statement “...*when-ever we want developers to participate in the ecosystem by using our APIs or to implement google technologies etc. we have teams that are especially focused on that and we call them developer advocates*”, implies that Google’s evangelism strategies are now so sophisticated that they have customized training solutions for both individual developers and c-suite technical managers (Rsp1: 8).

Furthermore, Rsp1 (48), Rsp5 (66) and Rsp6 (37) highlight contractual agreements as a critical tool for relationship management to establish what the developer can expect from the system and what the system expects from the developers. Rsp5 highlights an important consideration when describes how big accounts, who often want customized solutions and interactions with the system, are provided specific contracts meanwhile standardized terms and conditions suffices for smaller actors. As different businesses have different scales of developers working in their systems, there is often a need for different types of contracts (Helm, Holland, & Gangopadhyay., 1990). Other social boundary resources consist of developer advocates, account managers, product consultants and other evangelists of core platform features who act as important liaisons in the attracting and engaging developers (Rsp1:12). This is in line with Ghazawneh and Henfridsson (2010) process perspective of platform governance where they acknowledge the need further evolution and design of new boundary resources. Such evolution can be observed in Rsp3’s (32) example of the direct plug and play configurable applications given to third-party developers, allowing hosts to target developers that do not write code to integrate with the core platform. This draws similarities with the concepts of IFTTT (If This Then That) which through simple user interfaces allows users to directly manipulate the application in what they call applets to their own automation needs riding on the core service (Martin and Finnegan, 2018; Diller, Shedroff, & Rhea, 2005).

5.2.2 Critical Mass

Even with all interfacing mechanisms set in place, a platform still needs enough producers to drive demand and offset network effects. Known as critical mass, this remains an issue that concerns all organizations that aim to partake in third-party development (Altman and Tripsas, 2014). Critical mass determines the relative advantage as perceived by the users in using the services as initially provided by the platform host (Rogers, 2010). The launch phase of a platform is a delicate state where those who succeed manage to drive demand quickly (Evans and Schmalensee, 2010; Rsp3:28). Several of the respondents supported notions made by Evans (2008), claiming that size, reach and financial stability are essential aspects of to reach critical mass (Rsp1; Rsp2). Thus, critical mass is less of an issue for actors with “deep pockets” and a global audience, since they can rely on their brand to on-board users in whatever pace they see fit (Rsp1:22). Being a financial services vendor, MasterCard do for an example conduct a long vetting process of third parties, but as long as stakeholders can see future profit slow adoption is seemingly a non-issue (Rsp6:7). Rsp2 (35) examples of how companies like Salesforce.com have perfected their launching new platforms by strategized use of logistics, advertising and marketing are almost identical to those mentioned by Hart and Tzokas (2000), and indicate that the experienced market actors ignite critical mass by using strategies similar to those of new product launches.

However, according to Rsp2 (41) even smaller actors starting off a developer platform rarely face Evans's (2009) dilemma of simultaneous entry. Strategic partnerships with already established actors helps enlarge the addressable market and are attractive for all parties as they help create a one-stop shop for the consumer (Rsp3:14; Rsp1:22). Simply having a product is not enough but the strategies put in place to communicate its capabilities to the potential adopters to enhance its observability is crucial in attaining critical mass (Eaton et al., 2015; Rogers, 2010). Rsp2 (13) refers to this as developer marketing where platform owners develop mechanisms to maximize developer reach. Rsp4 (30) emphasizes this in his statement "...you have to sell it as a product. if you are releasing an app into the market...". Nevertheless, findings align with Altman and Tripsas (2014) notions of expenditure remaining as a key issue in the strife for critical mass following platform launch. Rsp3 (40) highlights the often-overlooked cost component of early talent acquisition. As platform design and maintenance often requires specialist skills, lacking competence early in the process could block progress and require important development-dollars to be spent on recruitment (Rsp3:40).

5.2.3 Organizational Identity and Differentiation

The shift from a product to a platform-based business model is not just about operational changes but also impacts how an organization views itself in what (Altman & Tripsas, 2014) refer to as organizational identity to align with the new business approach. This is supported by the findings, where for an example Google's Android propelled the identity of the organization as a platform company (Rsp2:39). (Zhu & Furr, 2016) however argue that a company's main attraction should always be centered on its main product, a factor that most respondents seem to differ with. Several respondents claim that in the highly competitive and versatile technology market, the strive for profitability and market leadership creates a need to constantly innovate and diversify existing business models to establish new value propositions that third parties can extend (Rsp1:28; Rsp3:14; Rsp4:20). This strategy however differs with the size and maturity of the organization. Younger entrants tend to claim identity on a particular product market, as seen in many upcoming platforms, meanwhile incumbents that leverage their existing market leadership to carve out a different niche (Glynn & Abzug, 2002).

Organizational identity is thus particularly influenced by the differentiation mechanisms that come to play. This is with respect to the heterogeneity in ecosystem agents who consume a platform's intrinsic features driven by necessity or quality offerings (Wan et al., 2017). It is exemplified when a firm seeks to provide similar or associated services bundled onto their core product, or when forming strategic partnerships with other firms to deliver seamless services to end users through co-competition (Rsp3:22; Rsp4:14; Cabrera, 2014). Co-competition is a strategy whereby one cooperates with competitors to derive maximum value while protecting one's interests. This is also emphasized by Rsp4 (8)"...at no given point will you be able to for example be sustainable if you are not continuously integrating with other businesses". Harvard professors Adam Brandenburger and Barry Nalebuff in their book titled *co-opetition* highlight the depreciation of the "winner takes all" mentality asserting the inevitable dynamics of businesses being dependent on the success of others (Mankevich, 2014). But given all such differentiation and competitive mechanisms, the heuristics that guide organizational action and interpretation by the external environment should not be corrupted by the shift to platform business models (Kogut & Zander, 1996). A conclusion drawn by most respondents therefore is that this factor is dependent on market position, competitive motivation, size of the firm, type of business model and the degree of openness (Altman & Tripsas, 2014; Rsp1:48; Rsp6:19).

5.3 Governance and Control

After understanding developers and having appropriate platform design strategies in place, the last phase is to establish policies for proper implementation and monitoring of the business model actualization.

5.3.1 Platform Rules and Communication Structures

Platform rules are of utmost importance in this context, implying the manner in which the platform owner administers the interaction processes with external innovators in order to comply with internally set standards and propel product innovation (Perrons, 2009). To effectively implement this, findings indicate the importance of doing market research to understand user preferences (Rsp1:24; Rsp2:53). This aligns with the previous discussion on developer marketing that in this context places the third-party developer as the targeted user. Staudenmayer et al. (2005) express the new type of dynamics that arise when the platform owner has to take into consideration the ecosystem of actors that could be impacted by major changes in the core product.

However, findings indicate that companies today include the aspect of backward and forward compatibility to ensure that any innovative improvements on the core application and subsequent API does not require significant changes from the side of the third-party developers (Rsp2:55; Rsp4:46; Rsp6:43). Rsp5 (62) showcases this in his statement “*we have very minimal changes that end up affecting external customers, but the reason why is that we make sure that regardless of what we do on our end, the developers never need to change how they integrate to our system*”. Accordingly, Weinreich, Ziebermayr, and Draheim (2007) emphasize this as a way to assure the customer that a product will not become obsolete even with the unavoidable service evolution. They define backward compatibility as designing to allow for a product to be compatible with previous versions of itself while forward compatibility allows for compatibility with future versions of a product. The concept of conflict free product versioning in itself is also a consideration highlighted by Rsp6 (43) as similarly done by Clever, Holler, Püster, and Shitkova (2013) to prevent loss of information and enhance platform adoption in accordance to the diffusion of innovation factors by (Rogers, 2010).

In the theoretical review, the aspect of designing pricing rules and models is portrayed as particularly important. However, findings do not highlight this as a major consideration and instead weighs more on the type of business relationship as defined by the platform owner and dictated to by the type of business model (Rsp3:36; Rsp6:7). A clear distinction is showcased between permission-less and negotiated platform access (Rsp6:7; Parker et al., 2017). Negotiated access describes an arms-length, contractually oriented relationship where the external innovators have competitive relations among one another (Boudreau & Lakhani, 2009). Here, the platform can capture value through direct licensing or contracting with external actors that increases profits through direct sales (Parker et al., 2017). As seen in the findings, such companies often require vetting of potential partners, particularly evident in financial services where security overrides all desired benefits of a more open approach (Rsp6:33). This is different from allowing creative freedom where any external developers can experiment on novel solutions and incorporate them into the core product with minimal governance, as in the case of building apps for Android or Apple (Rsp6:7; Leijon et al., 2017). This approach eventually builds an enhanced demand for the core platform through multi sided network effects (Top et al., 2011).

Top et al, (2011) argue that negotiated access approaches have in many instances risen to monopolies and more profit generating firms as evident in the likes of Mastercard and Visa, contrary to arguments implied by Rogers (2010) that these strict on boarding measures could increase the complexity in adopting the subject platform. But a more recent trend is the creative merge in both negotiated and permission-less approaches as once again demonstrated by Apple's hardware and Appstore strategies to derive maximum benefits (Parker et al., 2014). Similarly, Rsp1 (46) highlights the issue of policy compliance as do Manner et al. (2013) who back this argument by summarizing governance policy considerations of standardization, distribution, quality of service, security, finance and legalities.

Governance design choices involve decisions on management practices, leadership and ownership (Nocke, Peitz, & Stahl, 2007; Tiwana et al., 2010). To a high extent, they rely on the considerations for the above-mentioned design rules that are enforced by the platform owners (Boudreau & Hagiu, 2009). Teachings from open source software imply having a community of developers who work voluntarily on a product while coordinative creative relationships among them (Feller & Fitzgerald, 2002). Similarly, findings show the importance of engaging developers around a platform's core product by providing avenues that facilitate knowledge sharing and a sense of belonging through ecosystem or developer networks (Rsp1:54). O'Mahony and Ferraro (2007) support this as an effective strategy by referring to it as a concept of self-governing communities that blend democratic and bureaucratic mechanisms to allow for creative experimentation among members. This also goes a long way to encourage diffusion of the platform's innovation through complexity reduction for third-party developers (Rogers, 2010). In this context communication becomes very important often in form of lean channels complemented by occasional events and meetups (Kouzmin & Korac-Kakabadse, 2000). This ensures a decentralized level of control for the knowledge resources in a platform ecosystem (Yoo et al., 2008).

5.3.2 *Balancing Creative Freedom and Control*

Platform environments also comprise of co-dependent relationships between the hosts and complementors that often result in a push and pull regarding encouraging creative freedom of external parties versus safeguarding competitive advantage (Ghawzaneh and Henfridsson, 2013). Findings dictate that there is no one right way to do this and is often dependent on the type of business model or how a company chooses to conduct business (Rsp2:47; Rsp6:41). According to Laffan (2011), the type of governance is what determines the difference between a closed and an open project. She gives an example of Android which is arguably one of the most successful platforms yet ranked as the most closed project in the Open Governance Index. This aligns with Rsp5's (30) statement that "...you can still have open source (systems) and still make business for yourself or the company...". Similarly, at this point, strategic partnerships and alliances become of importance both within and without the firm to lure cooperation in innovative efforts resulting in superior products (Rsp1:36; Hagedoorn, 1993). This also includes the decision to ensure compatibility with other key platform players to foster innovation versus offering a stand-alone product in is referred to as multi-homing (Boudreau, 2010). Strategic partnerships however could result in frictions in terms of quality and standards between organizations. This often gives rise to vertical integration as a tool for extending market power, improving coordination and avoiding contractual conflicts (Bresnahan & Levin, 2012).

Another factor that emerges from the findings concerns the important differentiation between outsourcing services versus extending a digital business model. It is clear that both strategies

when implemented well would positively impact a business (Rsp1:56; Rsp4:26). Holcomb and Hitt (2007) however emphasize that outsourcing is mainly undertaken to increase the efficiency and cost savings of a firm in a resource-based view perspective. This is contrary to leveraging external innovators in order to reach more customers through direct or indirect network effects (Zhu & Iansiti, 2012). Hence as Rsp4 (44) states, “...you will find situations where you give something out as an api and you are expecting people to build ‘ferraris’ with it and they end up building ‘Lamborghinis’...”, third parties have the potential of transforming a product ahead of what its initial use case was and therefore propelling it to new territories (Rsp4:44; Boudreau and Lakhani, 2014).

5.4 Enriched Research Framework

This table is presented as an enhanced version of the research framework presented in Table 2.1, enriched with empirical findings and subsequent discussion. When summarizing the factors, it becomes clear that both themes and sub-themes derived from literature are in some way addressed by all respondents, indicating a certain generalizability. When looking at specific key considerations however, several important additions to the original research framework has been identified. The considerations which have not been explicitly addressed by the presented literature in Table 5.1 are highlighted as “NEW”. Additionally, the bullet points under each consideration gives a more detailed account of its properties and indicates how it may be actioned. For further explanation, the column to the far right summarizes the ways in which considerations aim to drive diffusion among third-party developers, based on the five factors of diffusion of innovation theory as described by Rogers (2010). It is observable that the five DOI factors are catered to differently in separate phases, with some spanning over the entire process of establishment, meanwhile others are especially prominent in specific stages.

Furthermore, it is important to restate that the themes serve as phases, where each consideration lays the foundation of considerations for the following phase. The phases do because of this have several overlapping properties, even though the properties of the consideration may evolve as the organization moves closer to platform establishment. When compiling the factors, it becomes clear that some considerations stands out as being influential over several phases. Strategic partnerships are for example a seemingly impactful consideration over the entire process.

Table 5.1: Summary of key factors for consideration in platform establishment

Theme	Sub-theme	Key Considerations	Effect on diffusion rate
Understanding developers	Developer motivations	Addressable market (NEW) <ul style="list-style-type: none"> The addressable market concerns the market size available to developers 	The addressable market of developers can be depicted as the main indicator of relative advantage and primary motivation for developers.
		Strategic partnerships <ul style="list-style-type: none"> Interoperability with other leading technologies or platforms Enlarging the addressable 	Is in this stage, tightly intertwined with addressable market. Drives relative advantage and compatibility since innovations can be

		market	used in a larger ecosystem
		<p>Developer Support</p> <ul style="list-style-type: none"> • Soft sells • Online forums such as blogs, personalized consultations 	Decreases the technical complexity of working on the platform.
		<p>Education initiatives</p> <ul style="list-style-type: none"> • Hands-on labs • Product documentation, Tutorials • Seminars and Workshops 	All education initiatives decrease the technical complexity of working on the platform.
		<p>Brand equity (NEW)</p> <ul style="list-style-type: none"> • Product certifications • Brand association 	Being associated with a strong brand increases the relative advantage of the platform as expertise of systems with high diffusion is desirable.
		<p>Direct financial incentives (NEW)</p> <ul style="list-style-type: none"> • Rarely offered on permission-less development platforms • Important in negotiated relationships 	Financial incentives are discussed by previous literature, but not from a permission less versus negotiated perspective. This is important to highlight, as monetary rewards strengthen relative advantage in negotiated settings, but may weaken it in permission less contexts.
		<p>Familiarity with software stack (NEW)</p> <ul style="list-style-type: none"> • Developers tend to have emotional connections to the stacks they started off with • Language agnosticism 	Increases trialability and compatibility of the platform since these considerations makes it generally easier to experiment with APIs meanwhile decreasing complexity for a wider developer audience.
		<p>Usability and ease of integration</p> <ul style="list-style-type: none"> • User experience • Technical complexity 	Aims to increases compatibility and trialability meanwhile decreasing complexity.
		<p>System performance</p> <ul style="list-style-type: none"> • Security • Turnaround time • Load handling 	Increases trialability. No developer wants to work on a slow platform riddled with bugs.

Platform design	Boundary Resources	<p>APIs and SDKs</p> <ul style="list-style-type: none"> Seamless design <p>Personalized assistance (NEW)</p> <ul style="list-style-type: none"> Developer advocates Product Consultants Student Ambassadors <p>Contractual agreements</p> <ul style="list-style-type: none"> Customized contracts Terms and conditions/NDAs <p>Direct applications (NEW)</p> <ul style="list-style-type: none"> Plug and play capability 	<p>APIs and SDKs determine the compatibility of the platform while increases observability since results are demonstrated on-site. Both traditional and social boundary resources helps to decrease complexity of integrating with the platform by clarifying what resources are available to developers and what conduct is expected on the platform.</p>
	Critical Mass	<p>Launch strategy</p> <ul style="list-style-type: none"> First mover advantage Launch like a product <p>Developer marketing techniques (NEW)</p> <ul style="list-style-type: none"> Early access programs Developer segmentation Product bundling <p>Strategic partnerships</p> <ul style="list-style-type: none"> Allows for tapping into the userbases of partners <p>Talent acquisition (NEW)</p> <ul style="list-style-type: none"> Avoid recruitment during launch phase 	<p>Besides being essential for its existence, critical mass considerations like launch strategy or developer marketing techniques strengthen the observability of the platform by showcasing its capabilities. If there is not enough producers and consumers, developers will leave for other ecosystems.</p>
	Organizational identity and differentiation	<p>Strategic ecosystem alignment</p> <ul style="list-style-type: none"> Coopetition <p>Product diversification</p> <ul style="list-style-type: none"> Profit maximization Thought leadership 	<p>Despite not having a direct relation to platform diffusion, organizational considerations are crucial for a creating a sustainable platform model who has a strong relative advantage.</p>

		<ul style="list-style-type: none"> ● Scale and management structures 	
Governance and Control	Platform Rules	<p>Consistency</p> <ul style="list-style-type: none"> ● Backward/forward compatibility ● Versioning ● User research <p>Relationship management and resource use</p> <ul style="list-style-type: none"> ● Permission-less versus negotiated platform access 	Platform rules decrease complexity meanwhile having the possibility to increase compatibility for the developers that choose to integrate with the platform.
	Balancing creative freedom and control	<p>Degree of openness</p> <ul style="list-style-type: none"> ● Risks versus opportunities ● Sandbox environments <p>Control through compatibility, strategic partnerships and policy compliance</p> <ul style="list-style-type: none"> ● Consider vertical integration ● Cross departmental inclusion ● Regional laws e.g. GDPR <p>Outsourcing versus digital business model extension</p> <ul style="list-style-type: none"> ● Cut costs versus allow enhanced product innovation 	Balancing the aspects of control versus creative freedom ensures stability of the platform, which in turn contributes to its relative advantage. Additionally, Sandboxes and staging environments are practical ways to facilitate openness meanwhile increasing trialability of available resources and functionality.
	Communication structures and leadership	<p>Local developer ecosystem networks</p> <ul style="list-style-type: none"> ● Social meetups and events <p>Technology channels</p> <ul style="list-style-type: none"> ● Real-time community forums ● Mailing lists 	Means of communication strengthens observability and trialability since experiments can be shared and discussed. This will also decrease complexity by extension.

6 Conclusion

This chapter summarizes the research study by highlighting prominent findings. The first paragraph describes how the study has answered the research question and fulfilled the purpose stated in chapter one. The following paragraphs present the key findings and concludes that there is a need for more in-depth studies of developer motivations and behaviour, as this is still a rather unexplored but increasingly important group of stakeholders

6.1 Research Question and Purpose

Much points to the fact that O’Grady’s (2013) naming of developers as the new ‘kingmakers’ is an understatement. Using external innovation to extend your business model has become a significantly impactful way in which companies differentiate their product portfolios and penetrate new markets. Establishing a platform for third-party development is however a process not entirely free from risk, as can be seen by failed attempts by actors with significant market share. Thus, it has come to be of importance to understand the different considerations that must be made to attract and retain developers in the ecosystem.

This study has built on the notion that some considerations need to be made regardless of the company business model. Understanding developers, platform design and governance and control are stages all hosts needs to go through when establishing their developer environment. Perceptiveness of these considerations thus influence the success of the consequent developer platform. This helps to explain how platform entrants have been able to triumph over large incumbents, and the fact that platform business models are being diffused over an increasing number of industries. Third-party developers are still fairly ambiguous assets and there is a lack of research outlining what considerations that have to be made in order to attract and retain developers to hosted platforms or ecosystems. This has led us to the following research question:

Which factors are important in a platform ecosystem establishment - that attracts and retains third-party developers?

The purpose of this study was to create a list of factors that c-suite managers can consider when aiming to establish a platform ecosystem that attracts and retains developers. Drawing inspiration from the process perspective of previous studies in the field, considerations have been grouped based on a theoretical framework consisting of three major themes, all corresponding to a phase of platform model establishment. Each of these have in turn been explored through a qualitative study where six interviews have been conducted with experts from different platform models, all with significant experience in managing or researching third-party development. The identified considerations have in turn been analysed through the lens of diffusion of innovations theory where observability, compatibility, relative advantage, testability and complexity are used to explain how the consideration supports platform diffusion among developers.

The study has contributed to the field of information systems by providing a high-level overview of key considerations for platform establishment, as presented in Table 4. It has done so by compiling, validating and adding to the results of previous studies in the field. New additions

have been highlighted and explained, in order to emphasize the theoretical contribution and encourage further investigation. Additionally, the study has not only resulted in a comprehensive list of factors, but also analyses how these factors drive adoption among developers by applying DOI theory. Using DOI theory as a tool for analysis provides a new and deeper insight to our understanding of successful platform business models, and how considerations impact platform diffusion.

Findings have been presented in structure that highlights the process of platform establishment and facilitates comprehension of its different phases and components. Thus, the findings presented in Table 4 answers the research question and fulfils the purpose of the study by outlining and explaining the most prominent considerations for establishing a platform ecosystem that both attracts and retains developers and is characterized by a high diffusion rate.

6.2 Key Findings

Despite the vastly different business models, it is evident that all responses gravitated towards the themes and sub themes outlined in the research framework. This supports the presumed generalizability of the process perspective and considerations needed to establish a platform ecosystem for third-party developers. Subsequently, all the five factors of DOI-theory were either explicitly or implicitly addressed by respondents, showing that platform hosts approach developers as prospective adopters of new technology. Furthermore, there were a number of findings that had not been explicitly clarified by the investigated literature. Interestingly, a clear majority of these were discovered in the first phase of understanding developers, indicating that this area would benefit from more exploration. Findings important to highlight lists as follows;

Addressable market

Congeniality of answers regarding what developer motivations are of most importance highlight addressable market as a key finding. Although not explicitly stated by literature, this motivation is seemingly core to third-party developers. For hosts, knowing how to quickly enlarge the addressable market is a crucial consideration.

Technology brand equity

Brand equity is seemingly a fairly unexplored but important aspect of developer motivations. Findings show some of the biggest actors in permission-less third-party development build their value proposition around brand association and certifications, rather than direct monetary incentives. As a strong brand usually often imply a large addressable market, there is probably a correlation between these two factors.

Permission-less vs negotiated platform access

Both permission-less and negotiated platform access posed interesting research objects, especially in the establishment of platform rules for governance and control. A clear variation between the two is the degree of openness that closely ties to the type of digital business model. While respondents emphasised the importance of financial incentives in negotiated contexts, monetization was a far less concern in permission-less developer relationships. Permission-less developers seem to value other more ‘soft’ incentives higher than direct financial compensation. The discrepancy between findings and literature therefore indicates that motivating developers

is a complex and highly contextual consideration that depends on the nature of platform and corresponding business model.

Strategic partnerships

The study has found that strategic partnerships is the one actionable considerations that impacts all phases of platform establishment. Thus, Strategic partnerships are concluded as a necessity when pursuing a platform business model for external innovation. Findings show that strategic partnerships drive developer motivation as it is what really opens up the addressable market. It is the primary consideration of design as it enables the onboarding required for critical mass and it allows for comprehensive control and governance when the platform is up and running.

6.3 Implications for Future Research

This study concludes that every company seeking to establish a platform ecosystem, given the current convergence in enterprise and consumer technologies, considers a number of key components for establishing an ecosystem for third-party developers. Developer motivations, platform design, governance and control all come with its considerations that will determine the level of diffusion of the platform. We believe that these considerations could serve as an explorative starting point for research, where future more explanatory studies could investigate the relative strength of relationships that each consideration has with platform diffusion rate. Additionally, this research has only applied the impacting five factors from DOI theory. It would be interesting for future studies to include the DOI grouping of different kind of adopters in the context of third-party development, and how to best serve each of these categories. We also call for more in-depth studies of developer motivations as this is still seems to be a relatively unexplored aspect of platform research. For example, the relationship addressable market and brand equity is undoubtedly interrelated and significant to developers in terms of motivation and would benefit from more focused studies. Lastly, ecosystems are complex but build on partnerships and synergies between organizations, technologies and platforms. Future research should investigate how firms can strategize partnerships for optimal third-party development.

For practical implications, it should be acknowledged that different business models and industries may come with specific considerations which are outside the scope of this paper. However, we hope that our findings can serve as foundational guidelines for those aiming to extend their business models by using third-party developers. This research has supported our previous notion of platform-based development being the future of innovation. The respondents similarly agree that all organizations must in time open up their development processes in some way and capture the immense value available through third-party development.

Appendices

Appendix 1: Interview Guide

Introduction:

- General introductions of both parties to establish rapport
- Confirm if ok to record the interview
- Explain our research motivation/statement and purpose
- Discuss and confirm briefly the interviewees' profile due to prior knowledge of their background and current role
- Clarify any other concerns the respondent may have

Main Interview:

General:

- What is your view on the future of digital business models, do you think that platform-based ecosystems will keep trending?

Theme 1: Understanding Developers

- How would you say your organization or one of your products uses third-party developers to create and capture value?

Category 1: Developer motivations

- What would you say are the main developer motivations and how do you think developers prioritize these motivations, like the access to educational tools, support, documentation and monetary incentives?
- How do you think the market position of the company based on its product differentiation impacts a developers choice of platform?
- In what ways do you reach out or evangelize to the developer community or those in B2B contexts (e.g events)?
- What strategies do you have in place to lock in developers or third-party development onto your platform? (how to increase switching costs)
- What do you think are the most important and generalizable motivations for developers when choosing one platform over another? Is there any that gives a particularly strong relative advantage to others?

Theme 2: Platform Design

Category 1: Boundary resources

- How do platform hosts strategize their distribution of SDKs and APIs? What different aspects do they take into consideration in such design (e.g distribution channels)?

- How do you optimize the user experience (interface design and usability aspects) of third-party developers on your platform?
- What resources beyond SDKs and APIs do platform hosts use to engage and retain developers? Why are these important?

Category 2: Critical mass

- There are a number of examples of platforms that failed due to being unable to reach a critical mass of users in time (Google video is one), how do you think organizations strategizes the launch of the platform in order to reach critical mass?
- The launch is a crucial phase for the future success of a platform, what do you think is especially important for organizations to keep in mind during this stage?

Category 3: Organizational identity

- One important aspect of organizational work is building an organizational identity (who are we as an organization, what do we do, how do we do it). Do you think the identity is affected by applying a platform-based business model? If yes, how?
- Do you think that a company should remain focused on their core-business when pursuing a platform business model? Or is it more important to switch focus to create a healthy developer ecosystem?

Theme 3: Governance & Control

Category 1: Platform rules

- When aiming to launch a platform that extends the business model, we presume it is important to take competitors into consideration, how do strategize for competitive advantage?
- Does the transition to a platform-based business imply a change in how the organization works with the development processes of their core business? If yes, how?
- What rules of engagement do you set up with external developers in order to build power and maintain trust (i.e pricing models)?
- How do you ensure other platforms do not infringe on your product in context of the architectural design for instance?

Category 2: The balancing act

- Do you think that collaborative markets have served their course? Are proprietary going to keep the way forward or will it be a place for collaborative markets in the future?
- We understand that exercising control over a platform while facilitating creative freedom of third-party development is of conflicting nature. How do companies handle this balance?

Category 3: Teachings from OSS

- How important is “lean communication” in platform ecosystems and how do organizations work to facilitate both platform-creator interactions and creator-creator interactions on their environments?

Closing questions:

- What factors would you rank as most important in establishing a platform product that captures and motivates third-party development (given the dev marketing, architectural and governance strategies discussed) ?
- Is there anything else you feel we have left out but is important for this discussion?

Thank interviewee and note any interest for follow up questions or research outcomes

Appendix 2: Interview Transcript [Rsp1]

Rsp1 = Respondent 1

Line	Person	Content
1	Researcher:	Your idea of a platform business model seems to align very well with what we have studied. Firstly, can we record this interview?
2	Rsp1:	That's fine
3	Researcher:	Briefly, what is your profile at Google?
4	Rsp1:	Cool, so I work in the sales organisation, large customer sales. blue chip companies which is companies that have over 10 million dollars in market value. what I do in the sales organisation is I am basically a strategic value ad to the sales team which means that I lead mobile assets. My official title is mobile web and measurement, what that really means is that I lead UX. Helping advertisers to have really great mobile sites and websites, because if an advertisers website is good, that means that conversion will be better and that the advertiser will spend more on Google and the other aspect is measurement which means i work with advertisers on marketing analytics so think of like retailers who want to measure online to offline who want measure people who looked at an ad and walked into a store. I do a lot of statistical experiments using R so you know let's say that you're an advertiser and you have an hypothesis such as one advertising channel performance better than another. which means that we have to find a way to measure whether it was a TV or some other channel that drove this X number of sales that you see in business. sort of like causality type of experiments. We do this to help advertisers fulfill their revenue commitment and then the other aspect is just general measurement like how do you know that a user clicked on a facebook ad and clicked on a google ad which led that to that they saw a TV ad and eventually come to purchase on your site. How do you measure all those multi touch points to really understand sales funnel that they used or interacted with before they converted
5	Researcher:	Interesting, seems pretty techy for someone who is
6	Rsp1:	Haha no as long as you know how to do R, python, basic knowledge around javascript and obviously solid knowledge of statistics. that's all you need to be a fully measurements specialist at Google. My work is really driving incremental revenue. An advertiser spends a million dollars and if a sales manager wants to grow that to two million dollars, my work is really about, how do we do that?

7	Researcher:	If we're gonna get into our topic now it's three general themes that we would like to talk to you about which we have identified, firstly it's developer marketing/motivations which literature claims being an important consideration. Secondly is platform design and third is governance and control. We're gonna start of directly with the first one, how do you think developers prioritize motivations, access to educational tools, support, documentation, monetary incentives is that any motivations that stand out among the others when you target developers?
8	Rsp1:	It's a very interesting question. how we usually go about it, we know that Google, Apple and Microsoft are top leaders in their industry right, when it comes to technologies that companies embrace and should adopt. Given that, as a developer, you want to associate yourself with the best of all when it comes to technology, as a developer you want to be able to call yourself a consultant because if you're working for like and SME, in most cases they want to develop android application or an iOS application etc. if you think about the vendors of the market who actually own viable infrastructure, it's google, Apple and microsoft whoever else that is in the market. There is like a brand equity aspect to it that motivates developer that wants to be associated with that brand. To market themselves as consultants that offers this type of expertise. Going back to the initial point that I said about producers and consumers. If you are to think about the android framework only, there are so many producers/companies that are building android apps and developers who are building those apps as well, we're considering them as producers etc. The addressable market is significantly big, what google as a company focuses on is curing that addressable market and at the same time also incentivizing and educating them too through marketing programs which we call partner programs. we don't give monetary incentives, the incentive we give in most cases is certification. you are able to have a certificate that means that you can offers this kind of expertise in for an example android to companies that are seeking that type of skill. To sum up, there's a brand equity aspect associated to these developer motivations. The importance lies in become "google developers" or whatever that is.
9	Researcher:	It sounds like has a lot to do with market position of the platform?
10	Rsp1:	Precisely, the market position of the platform has a lot to do with this right. If you are a company starting today who are selling a thing that is incredibly hot. To get developers to be interesting in your product, to start developing for your product. It's not as easy, the market position sort of dictates that
11	Researcher:	So you at Google, what do you do to reach out or evangelize to the developer community? Or in B2B contexts as well. We have professor that claims that CIOs and CTOs, they don't make the decision these days on what platform to develop on, they don't have the insight or the time so usually they go direct to frontline developers, i.e. there is an importance for platform owners to actually reach out to the developer community as well. I know you talked now about certification for example. Do google have any other targeting events or marketing?

12	Rsp1:	<p>That's a very interesting question right so we have multiple merits with how we go about this. As you know google's business is structured as this: the sales team, the marketing teams they do very different things but all of them are geared towards one common objective, it is as simple as encouraging consumers and producers to participate in our platforms and products. to get advertisers to continue advertising. The thing is, we have developer teams and we call them developer advocates whenever Google comes up with any type of technology, the new thing now in the market is mobile, everywhere in the world not just in Africa and Google has really capitalized on that in the sense that we develop technologies that is going to shape the future of mobile where people are actually moving from developing native applications to actually create mobile sites that have functionalities that have functionalities that a native application would have, we call them progressive web apps etc. because if you have a native business model and you're creating an app then you are competing this space with instagram, snapchat, facebook etc. and no matter how much I like your brand the frequency of visibility would really matter if I'm gonna use your app or not that's why mobile site directions makes most sense. To answer your question, whenever we come up with any type of technology or whenever we want developers to participate in the ecosystem by using our APIs or to implement google technologies etc. we have teams that are especially focused on that and we call them developer advocates. It's a breadth and depth approach so. The breadth approach is where we leverage the developer advocates to do events, they are called hackathons. Given the market position that we have, say that we are doing a hackathon event in Stockholm and we would invite developers to come and we would teach them how to actually let say, improving the speed of a mobile site by doing javascript modification or implementing whatever type of technology they are pushing. we are actually helping developer advocates in doing this kind of things right and then the depth approach is where you actually have consultants sitting within the sales teams so like myself. While we do this on a consultative approach so we work with our managed clients. Not all companies out there are google customers and not all companies want to advertise, however companies that want to advertise with google, we have a special program for them where we have specialists or consultants sitting in the sales organizations that actually work with this companies appraise this google technology. that's sort of like the in depth approach where we provide... I do exactly the same thing as developer advocates do, the difference is that advocates focus more on a breadth approach with hackathons etc. but I'm focused on more of a client by client basis. Engaging with CTOs and CIOs, obviously a lot of these kind of projects would need to be funded by the CEO and funnel down the organization, but if you were to draw a power interest-creed you have a CIO who is someone who'd have very high interest and significant power on whether they are gonna want to adopt the technology or not and that's why you need to have like consulting skills to be able to get the mind of this CIO to review and validate your product and all that kind of stuff and because of the market position we have at google, a lot of these companies wants to embrace all of this technology and then the other thing that is really important for me to mention is as google's approach is very different from apples, so if you would interview an apple employee you would get different answers but we always try to back open source products.</p>
13	Researcher:	<p>To fill in on that, we have a question if you think that collaborative markets like open source project have served their course since we're seeing proprietary platforms trending more and more and is somewhat collaborative initiatives in that sense. Google must find a value in supporting open source, right?</p>

14	Rsp1:	Really all types of projects I have done, whether it's in the mobile webspace etc, it's not proprietary, it's open source but what we do is dedicating resources in terms of engineers to back up the technology and make it more efficient. For instance, in the web today this new technology that companies are working with called service worker. Service worker is going to replace native application so today mobile sites can look exactly like native application where you can add to homescreen and load exactly like a native application without opening an address bar, it will be able to work on offline and in flaky connections, you'll be able to do push notifications and all those kind of stuff. however, service worker is an open source technology but Google would have funds and marketing teams behind it supporting that kind of technology because it aligns with our business model and our main objective. we do a lot fo these types of things to support open source platforms. So even android as you know it is not proprietary, it's open source whereas apple's approach is slightly different, it's kind of like a closed ecosystem and it works, it's not like it doesn't work, it's just different. For us it's always very much about letting the ecosystem to participate.
15	Researcher:	Do you do anything specifically to lock in the developers to android, when they have started so they don't transition to competitors?
16	Rsp1:	Yes we do, very good question, and we do that by increasing our market share and market position. the more addressable market we have, the more you as a developer are locked in to the google ecosystem.
17	Researcher:	Okay, fair enough
18	Rsp1:	If you look at the speed of android versus apple, android has roughly 80 percent market share and that dictates a lot. You are thus more inclined to develop for android just because they are an addressable market. Because we also have consumer products right like google search, gmail, youtube and all this kind of stuff and what we always do is to get a lot more people to want to use google products so these are the consumers and once you are encouraging consumers to participate in the ecosystem, you are inherently pulling the producers towards participating in the ecosystem.
19	Researcher:	That's interesting. When you look at all these motivations we are talking about, having evangelists or hackathons and just propelling your market share position. what if you have a new product, something that is not been in the market like android for an example. you just launching for an example google wallet
20	Rsp1:	Google pay, yeah
21	Researcher:	Yeah, in that case, do you market differently? to developers?

22	Rsp1:	<p>Yes, all those new products and existing products.. The way you structure a market or innovation, it's usually a market that is mature or introducing the same product to a new market or you want to create a new product entirely for a new market. or create a new product for an existing market. To put this in an axis and draw a X and Y axis you sort of come to different conclusion in how you go about marketing these different situations. However, going back to answer your specific question. whenever we create a new product, we always tap into the existing ecosystem. So as an example, we know that people really are using android as operating system, google pay is just a product that taps into the android ecosystem. It's not necessarily a new or disruptive innovation it's just adjusting to the core innovation and taps into the existing user base that we already have and the way we go about this is... I'm trying to think of a good use-case, I'll get back to you on that! however, whenever we create a new product we just tap into existing ecosystems. we already use a number of google products and all that we'll create a new product that makes the entire google experience better. Google pay makes it easier for you to do payments right, by just tapping your phone instead of carrying around your cards. the first thing you learn is that google is a thought leader in technology so I believe in that product and if some company in africa created a similar product, no one would bother or they would have to spend so much money marketing so get market share. With google however, we already have that user trust, it makes a bit easier to market a suite of products. normally we would use certain growth hacking techniques, so growth hacking is an entirely different topic that you can read about in how tech companies that like large user bases and how they use growth hacking to do their marketing instead of them doing billboards or this kind of stuff. it's about how you use existing products and do things like bundling and they're so many of the, I can't really remember but it's slightly different. As an example, google pay has tapped into things like maps API so if you download google pay today and you walk into a store that accepts google pay, you most likely get a push notification, that tells you, android pay is accepted here. that's an example of growth hacking techniques where we are able to tell the context of the situation and where the user is and be able to tell them that they can actually use android pay there. that's just some example how we go about using existing ecosystems to grow user base of new products. But it's different ways in how we go about driving user adoption. If we go to Africa for an example, we don't have the market share like we do in Europe or in America. We have existing local companies that have existing companies that have better market share than we do and in such cases how can we support these partners in a way that aligns with our business model? in the, end we're also not an NGO.</p>
23	Researcher:	No, of course
24	Rsp1:	<p>For an example when we took google search to Africa, we had to do market research to understand how users actually use search in africa and we realized a lot of people are actually looking for jobs and that because of the economic situation and we were able to partner with a lot of job application sites, so that if you logged into google search and typed jobs or something like that that would actually be part of the google search ecosystem rather than just their own sites so it's where we actually integrate with the locals. lock mass on product to make sure that we're not taking an innovation that works in the US to africa and expect it to work the same. For instance, understanding context like cost of data is very high in africa which is because of the purchasing power. Someone spend 20 dollars buying data or they spend 20 dollars buying food for the family so it's more like, how do you create light versions of the existing products so that you can actually recruit users. there's a lot of contexts that goes into these</p>

25	Researcher:	<p>Alright, we're gonna move on to the next theme which is platform design and as I said previously, google is the incumbent. it's part of the GAFAs companies and probably the biggest one of them. So for these questions, you may have to think a little from your experience with google how would the approach be if you weren't such an enormous incumbent, we'll see if you understand what I mean. For an example, the first question is; one important aspect of organizational work is building an organizational identity, Google knows that if any organization I think, who we are as an organization, what do we do, how do we do it, what are our codes of conduct. Do you think that organizational identity is affected by applying or transitioning to a platform-based business model in comparison to a pipeline</p>
26	Rsp1:	<p>Can you repeat the question maybe in a different context?</p>
27	Researcher:	<p>Maybe you could say for an example that amazon is a retailer, you want to extend your market share and maybe even want to tap into different kinds of business models outside your own ecosystem. How does that really affect your core product? the core product for google is advertising I would say. Isn't it like 92 percent of revenue?</p>
28	Rsp1:	<p>You'd be surprised that advertising makes up like 60-70 percent of revenue. So that's an interesting question in the sense that we know that our core is advertising but we don't want to be left out ones something is happening around the world. Ideally, we tap into any kind of business models so as an example, google had invested so much in engineers and we had decent infrastructure with datacenters that were powering our ads however, over time google realized that people can actually tap into our data centers and our infrastructure. Why don't we enter into the cloud business model so that we can have dedicated resources that power our ad infrastructure and at the same time, because we we're already a thought leader in AI, machine learning etc, and we've proven that through google products such as search gmail etc and how we use AI to power products we can actually allow other businesses to tap into this thought leadership that we already have and that could probably drive incremental revenue to the core business and that's exactly the approach that google took. We just started cloud model two years ago but we're already at the top right we're like number three there, competing with microsoft and amazon. people who been in the cloud space for more than 5 years and what we did is that we extend things that we were doing creatively good, those were things like machine learning and we also acquired a few companies that were doing incredibly well in their area that we had a very strong investment interest in, but then we allowed other companies to tap into the existing infrastructure that we had, as an example spotify that is a swedish company is actually hosted at google cloud and the way has handled the saturation of music, understanding what type of music do you like and all those kind of stuff and being able to part personalized playlists for you it's all power by google's machine learning. So before we entered the cloud space, those kind of products like Machine learning, Ai etc. would only power google products and we really had to make people think beyond just developing these incredibly tools to power google products but also to empower the wider business ecosystem and that's subsequently really helps management because we sort of like diversify in the type of offerings. these are B2C elements, B2C is basically all like you using gmail on google search and the B2B element of it is more like be part of advertising but also partner products etc. and the point to that really is that when you reach saturation, you really need to think creatively on what are the next things that you can do and companies like google don't really use consulting companies to help them with their business model. We would consulting companies like McKinsey to come and help us manage the scale. We have 20000 employees and all of a sudden we are hiring 2000 employees right, that's why we need McKinsey, to help us manage that scale and processes, structures, systems hierarchy and all that kind of stuff to manage scale but we can't have McKinsey to come in and advice on how do you increase the market share on google search. Because only the founders of google would actually understand how to go about doing that so like even the way we go about tapping into new business models it's all done internally and we would never do that with externals or third parties such as consulting companies</p>

29	Researcher:	So it's actually more beneficial to tap into these other markets and to allow outside innovation to tap in. For an example, google cloud doesn't really dilute the identity of google as a company but rather makes it more diverse and appealing to more markets
30	Rsp1:	It doesn't, in the end, what's the central point that you probably want to address? you want your board of directors to be happy, you want your shareholders to be happy and how can you make your shareholders happy? how can you continue having that market thought leadership, you need to be open to business and you need to be open to tapping into new businesses models and what google did is we tapped into so many different business models and it became so difficult for google to manage and that's why google created alphabet company, so that each entity should actually be a company on its own. If you look at google cloud, even though it's branded as google, the CEO of google cloud does not report to the CEO of google, but the CEO of alphabet. If you look at deepmind for instance which is our AI company, deepmind is not a google company but an alphabet company and so on and so forth, there are like 20 of those kinds of companies that I don't know about. the way you structure your company research farm sort of like R&D and you'd have like your core business model and if you remember the matrix that I told you about there is always core innovation, adjustment innovation and disruptive innovation. If you are to draw a matrix for that it's like once the core innovation hits threshold you have to look into other stuff and once you got an adjustment innovation right you have to look into disruptive. Given though we're a data company and as you know data is the new dollar sign so given that you have data you can uncover so many business model and that's how google has managed to be so successful
31	Researcher:	There are a number of examples, of platforms data have failed due to not being able to reach what we critical mass. I think for an example google video before the acquisition of youtube did not for an example which caused the platform to implode. I was wondering, how do company strategize the launch of the platform to reach critical mass? maybe from a point of view when you wouldn't have the extreme brand recognition that you have, take for an example how did a platform company like Twitch do it? because you need some kind of plan right before launching your platform to get both consumers and producers?
32	Rsp1:	If you look at what company google is today, it is very difficult for google to continue innovating and when you reach that kind of scale you focus a lot more on acquisitions. We've acquired so many companies. Looking at the capital that google has and the manpower that google has and kind of the value infrastructure allows us to acquire easily and we've tried and failed on so many products, not one two three four or five but so many of them and it's just how these things are. we've proven thought leadership in like having like very creative product managers that are creating products that are having like companion users etc. and the way the platform scale works, you can't be doing that on daily basis even if you are like the most valuable company etc. and when you reach the size of google in most cases you just sit back and watch what is coming up around you which is an approach that facebook has taken by operating on instagram, oculus and all of these companies amazon have acquired so many companies and google or alphabet has acquired so many so many companies. you reach a point where there are other companies that get innovation right and you probably focus more on acquiring those companies. rather than spending resources on developing competing products yourself. The principle to project management means that ideally you you'd say as long as you solve a problem for the user your product will be successful, it's not black and white in that way. If that was the case anyone could go about starting problem solving techniques and start engaging users so yeah it's not black and white. Sometimes you just have to sit back and watch and acquire

33	Researcher:	Moving on to boundary resources and talking about SDKs and APIs, everytime you want someone to integrate into your product the obvious thing is to give them an API to just be able to just communicate with your platform so we'll just wondering are there any other aspects besides using SDKs and API in terms of architectural design that you consider in terms of exposing that product to developers that want to integrate it or is it just SDK, API and that's it. do you have some kind of distribution channel or?
34	Rsp1:	That's a good question which is super technical for me. I can try and find out about that, I can find out from a friend who is an engineer. The little I know is SDKs and APIs and that's how we expose our data to third-party developers because through that we'll be able to control that portal etc. I wouldn't know of other ways on how to integrate... the only thing that I've seen that does not necessarily use an API is google assistant. Google is trying to tap into the retail space as you know right. A study has been done that shows that the future of mobile is voice, people are gonna have voice based mobile phone and we're trying to be ahead of the curve we really invest in that in space by investing a lot in AI right. when you look at where AI is going to be let's say five years from now it's just going to transform the whole internet. We're integrating such things as customers... let me say like your supermarket and people shopping in your supermarket and users that already have google assistant can actually talk to the store and exactly where they need to go for products and as far as I remember it doesn't really exist like an API for that. Let me find out more about this and then I'll get back to you
35	Researcher:	Have you had any cases whereby people have tried to take advantage of google products for their own benefit?
36	Rsp1:	That's a very interesting question. I don't know if I have an exact answer but I can give you a very interesting use-case. when we came up with android. Android is open source, it's not really owned by google, we just power the ecosystem however, as you know the two main companies that are competing in the mobile space is Apple and Google, Apple has roughly 10 or 20 percent market share, however third parties, which we call partner developers such as Samsung, HTC, all these companies that develop for android.. because we don't own the hardware and we don't do the integration of the hardware and the software there's a lot of quality standards not really met. Hardware or devices that are not google standard and there was a lot of phones that were made in china that was shipped and since google doesn't control the quality standards of these integration, there was initially a lot of defects. so what google did in order to maybe compete with apple is to start creating its own line of hardware products and thus we have things like pixel or pixel book where we control end to end hardware and software integration. that means that we own the quality control of the integration and that helps us to create premium devices that are as good as apple devices.
37	Researcher:	I was wondering maybe on the same aspect since you work with UI and Ux, do you consider it from the point of the end user or the point of the third-party developer. In terms of usability

38	Rsp1:	Any time any day it's always from the end user, so what we have noticed is that a lot of our advertisers have really crappy websites that do not align with user design principles , performance and site speed not optimal, some websites take 30 seconds to load and the UI is not great, so we partner with third-party in such companies to do UX research on what should be a good mobile site for data, or travel or finance look like. We publish playbooks and a lot of research studies and that is how we get our advertisers to develop their mobile sites according to the best practices. We want to make sure that as a business if you are advertising google but you have a very high bounce rate, it means you as a business do not benefit because you are not getting conversion for sales and we as Google will not benefit as well so we provide this as a free ad on service for our large advertisers to help them develop best sites with best principles, and we have proven this even used AI to analyse millions of websites and we see that if a website has good speed there is a high likelihood of driving more conversions or engaging people a lot better. I could share those playbooks if you want.
39	Researcher:	Very interesting, thanks you, we did not know this as well. We have been through this a little bit with the evangelism questions and developer motivations but what resources beyond SDKs and APIs do platform businesses use to retain and engage developers like what do you do for them
40	Rsp1:	So we have certain incentives, we have an assortment of you becoming a google certified developer and we would pay for your accommodation and flights for you to attend Google events, so like the Google annual I/O, We would also recognize you as a google expert in particular market where google can be giving you some money to become a facilitator to train other developers in that particular market., and recommendations, like whenever we talk to businesses looking for developers who can build something like a progressive web app we would recommend those businesses to you. So we empower the developers. You will also be invited in alot of talks when google has to give a talk about any of its technologies, and a developer is google certified, google would sponsor you to become a facilitator . so they are technically monetary incentives but obviously because we can just give you money like this we sort of like use other ways to ensure it still aligns with google objectives
41	Researcher:	Do you still give out branded items?
42	Rsp1:	Yes we do give out a lot of schwags, and beyond developers we partner with educational institutions to not necessarily offer this as a curriculum but have ambassadors who can embrace Google technology
43	Researcher:	Alright, very nice,we are little conscious of time , so are you ready for the next theme? so we are wondering, we have a question on design rules and another on governance and control. I was wondering, the transition to become a platform, i presume you have to take into consideration this network of third-party developers, i am thinking of something like, suddenly we change the screen size of the handset phone we are selling and a whole network of third-party developers are being affected because of how they work with your APIs to develop . so is this implication correct, does being platform-based have an implication on how the org works with development processes basically if their own core business and how do you do at google to give your third parties a heads up , how do you support them to always stay compatible with your products.

44	Rsp1:	That is a tough question. I mean of course technology evolves right, to adapt the user. At the end of the day what matters the most to google is the end user. We do alot of research to actually see if any change particularly matters to the market We dont have top down type of decisions, we have an incredibly strong marketing team to help our managers make this type of decisions, but whenever we make this type of decisions we always align with all stakeholders internally, if it is the developer evangelists, or developer advocates who reach out to the people in charge of getting our content etc
45	Researcher:	Just a quick one, is there any specific rules you set in terms of how people relate with your product , is there anything that stands out in your mind or any particular rules important when working with Google? Basically relationship management with the developers
46	Rsp1:	I wouldn't really know that but i think the way we design our APIs for developers to use are already up to Google standards, so i am assuming, i have personally never been a google developer but something that really stands out to google is always the users privacy. That has always been at the core of any type of google product. so let's say for example you want to send data from your CRM to google analytics using a protocol API, Google does not allow you to send personal identifiable information such as an email address, so you will have to anonymize that., so that is just an example but more in a broader context there is always guidelines that google partnerships have to engage with its APIs etc
47	Researcher:	This kind of aligns with this governance question that we have that exercising control and having third-party partners do what you want them to do has sometimes a conflicting nature with creative freedom . and we understand there is a balancing act in this. how do you allow people to create as much as they want and at the same time ensure it is compatibles, does not infringe, its secure, and we were wondering this might also be outside your jurisdiction but how do you work actively to maintain that balance. How do you review what you should open and what you should close
48	Rsp1:	We have a team of like...you would be surprised the number of lawyers google has to ensure comply with all policies, we do alot of research and understanding about where technology is going for example with the GDPR coming in the EU, we had to update our privacy policy. If you were to think of the PM, someone who works with so many stakeholders, he is not the sole decision maker but he works with so many teams such as finance, legal, marketing etc that provide him with information that helps him to shape the product, and on top of that, if for instance if you have a PM in charge of APIs he would work with multiple teams to just design the API, he would get input from different teams such as legal council on what types of dimensions to expose to an API that may not infringe Google's data, so we have an incredible team that really understands all this things and work very close with the PM teams
49	Researcher:	Ok, alright, that sounded as thoroughly as we would expect, we are running on now and we have more questions, do you have more time now or do you need to go
50	Rsp1:	I can give you guys say like another 10 minutes?
51	Researcher:	Alright great, I think like we have talked about everything, we have talked about some of the rules of infringement, about how to balance control, maybe just one particular one, i know you have a way of communicating and keeping developers engaged , we had this GDG programs, are they still there?
52	Rsp1:	Yes , GDG ecosystem is very big and that's a group Google really values and this are the type of people we have been talking about since we started this conversation.

53	Researcher:	Yeah exactly
54	Rsp1:	This is a community of developers interested in developing for google products and embrace google technology purely on a voluntary basis, if you want to learn more about using google apis and such google would empower you with tools and events like hackathons etc Normally they meet on a quarterly or monthly basis and have a lead who understands more about google products and this person acts as the developer evangelist
55	Researcher:	Perfect, Just a mega broad one, taking the research question which is what factors organisations take into consideration when aiming to use third-party development to extend their business model. Do you feel there is anything in the general themes of developer marketing, platform design and control , governance and security. Is there anything beyond that that comes to your mind like we have missed or that is particularly important or are we in the right direction
56	Rsp1:	Yeah i can't think of anything beyond that , since its not really an area i have studied its just from me observing what happens. if i had done a little bit more research about this i would probably have shared a little bit more insights. So yeah i can't really think of any other things. And i think when you are thinking about this third-party development , you kinda need to draw a line between, like is it just about extending business models of apis because like when Sylvia told me about this, i thought initially you wanted to know if google uses third-party developers to help it develop its own products. But the answer to that is no. because for any platform business model or any type of business model, if something is really important to the business you don't outsource. Only outsource something not in the core business. For example google can use Oracle financial systems of HR systems , google can outsource that because its not part of its core business model, but in terms of actual consumer product, google would ever use a third-party to develop any of that as all is in-house, and also another example would be that google can never use a third-party to help understand how to expand its business model like the way some companies uses Mckinsey etc , so google would use them purely on, how do we manage scale . its simple as that. and it goes back to a lot of interesting studies you can read about companies that have used third-party developers and ended up even losing their own business model because this developers can create the exact same things. A very good case study is of this chinese company that would go to a company and tell them , hey we can actually do this for you at a cheaper cost, and they would go back again and say oh you guys are actually doing this, why don't we also do this for you guys at this cost, and the company kept on outsourcing and outsourcing and it got to a point where the company just focused more on branding the product and they were not producing anything and that way this other company that was doing all this for them was actually creating the same product and selling them to another company and re-branding the same products
57	Researcher:	What that is crazy
58	Rsp1:	Yeah that was a lesson and actually a case study that is always taught in many MBA classes . so if something is really important and the core of your business model, then don't outsource. But specifically on your question and your research is how do you extend a business model by leveraging the ecosystem, or apis and stuff. it's an incredible way to scale and encourage more producers to participate in the ecosystem.
59	Researcher:	Great. Thank you so much. much obliged really
60	Rsp1:	You are welcome. it has been great chatting with you guys at least i have also refreshed my mind

61	Researcher:	We will i guess send you some notes if you want them or
62	Rsp1:	Yeah sure feel free to send me some notes and if i remember something that i hadn't told you i will be more than happy to add on the notes
63	Researcher:	Awesome, have a good day. Thanks you so much. Bye
64	Rsp1:	Bye

Appendix 3: Interview Transcript [Rsp2]

Rsp2 = Respondent 2

Line	Person	Content
1	Rsp2:	Any organization that works with third-party developers has 4 divisions: you have the product which is how you build the platform with APIs and SDKs, you have developer relations they are the people who organize the hackathons and conferences, they write the sample code, they work with social media and write blogs, you have the people doing the marketing which includes how you present the product, who is it targeted for and how you generate demand for developers to download the SDK and so on and then the partnerships/BD which is building integration with others and with content or large tier one developers and other platforms.
2	Researcher:	I think we have all those aspects in the literature review embedded in our and it aligns pretty well with what we our looking at.
3	Rsp2:	What can I help you with?
4	Researcher:	Can we record this interview?
5	Rsp2:	Yeah
6	Researcher:	We're gonna have some specific questions around this themes, so firstly we are gonna start off with our first theme which is developer motivations or developer marketing. So our first question would be what are the main developer motivations and how do you think developer prioritize these motivations like access to educational tools, support, monetary incentives etc.
7	Rsp2:	Their motivations as people or the reasons they adopt the product?
8	Researcher:	The reason they adopt the product

9	Rsp2:	So let me start by addressing developer motivations in investing time in software development. A while back we published a model that identifies developer motivations and outcomes, what they are trying to achieve through software and how they measure success so these three factors and we saw that they are aligned and since then we've been measuring or identifying developers by identifying what outcomes they are trying to achieve by getting involved in software, developing students are mostly in it for fun and learning, someone who is a gun for hire or a hunter will be motivated by commercial and financial incentives and those working as employees will be usually motivated by introducing efficiency in their organization or using software to better reach customers or engaging customers through mobile apps or any other app. so very different types of developer in terms of what they try to achieve and what motivates them. This is one dimensions and you have then to consider that someone who is working professionally in the evening, they could be trying out an IoT project as a hobbyist. So in the morning they might be motivated by introducing efficiency in their organization, helping others get things done through software and in the evening they might be motivated by learning and achievement.
10	Researcher:	Fair enough, we understand that this has a lot to do with the profile of the developer. For our study we thought that we would delimit ourselves to professionals.
11	Rsp2:	Okay, another aspect that motivates developers are familiarity with existing software stacks. that is often talked about in what is called technology tribes so people who have experience and in some sense emotional connections towards technologies, for an example people which are very fond of the linux stack or people who have been using microsoft for a long time. In the old days there were usually people who have been brought up with Java and therefore have an affinity towards that technology stack. In terms of developer space, because you have to spend a lot of time getting familiar with a technology are victims to the nike effect which means that behavioral economics claim that you developer emotional connections to things that you have built yourself so you spend a lot of time working in c-sharp or java or javascript then you will start to develop an emotional connection and tend to identify with the tribe of that technology stack
12	Researcher:	Do you think that there's a brand equity aspect to this as well then?
13	Rsp2:	At SlashData we rank technology brands by technology satisfaction, adoption and engagement and they are public if you go on our blog we publish every six months an update. there is an another aspect that is probably the most important for anyone who is not a hobbyist, and that is the addressable market, in other words if a technology doesn't allow you to reach users, then it immediately loses its appeal, you can see that across history like nokia and microsoft didn't attract enough developers because they had a much lesser addressable market compared to android and iOS, you can see this with IoT that IoT didn't really pick up because they were too many devices and none of them were targeting hundred and millions of users, in the best case tens of millions and so you can see this with AR/VR now, it's been around for 6 years and is quite hyped but the hardware still needs to work so 80% of developers who are in VR are hobbyists and they're targeting android and iOS applications because that's the hardware people are having but you can't do really much especially AR with the small screen of a smartphone in front of your eye
14	Researcher:	In that aspect do you feel like the market position of the these companies are key considerations for developers

15	Rsp2:	Yeah, for an example google is along with microsoft and mozilla used most often by developers which means that more developers use them I think weekly than any other technology brand and that is because the technology footprint extends quite widely. If you look at satisfaction though you see game engines like Unity are the platforms with most satisfaction but they are not sued as often because games are more specialist. But the optimal model for building developer programs is unity more than anyone else
16	Researcher:	Just to go back to the list you had shared with us, do you think there are other ways in which companies evangelize to these developer communities.
17	Rsp2:	it's a lot
18	Researcher:	Any more prominent?
19	Rsp2:	Let me bring up a chart.. so the ones that are most popular are conferences and tradeshow, social media, it's having evangelists which are basically sales people in disguise, so an evangelist or advocate who will reach out to developers and see what they're building and see if they can help them build better applications through their technology but its a softsell so not a hardsell, you have meet ups, hackathons, competitions, incubator programs, I have 10 more
20	Researcher:	We think especially the parts about evangelists and capturing value that is created somewhere else is particularly interesting of course. you are helping someone with the intention of capturing value somewhere else. We have looked at ways in which companies reach out to developers, but how about keeping them locked in and keeping them engaged and make sure that they don't switch to other platforms
21	Rsp2:	It's very simple, in mobile companies are building mega-SDKs. As in SDKs that combine so much functionality in it, so that you end up using more and more and more of this from the same company and in the end of the day you end up using only microsoft or amazon or google technologies. Today, a good example is google's firebase and amazons AWS APIs so amazon has over a hundred different services and they all work very well together so the more you use them, the more you lock in and of course they work really well together.
22	Researcher:	So your expertise builds in your specific tools?
23	Rsp2:	Exactly
24	Researcher:	Yeah ok, and also your expertise builds in those specific tools and functionality right
25	Rsp2:	Yes exactly, previously there used to be some language locking as well for example microsoft had locked it in c# and oracle had Java and linux more C++ and C but now the modern platforms are very language agnostic , so if you look at the previous state of the developer nation published 18 months ago which showed that amazon for example was supporting all languages and the developers using it were coming from all language backgrounds. so the language locking is no longer present
26	Researcher:	Ok , and then lastly on this part, we see more and more proprietary platforms basically for example atlassian when they want to extend their business model buy allowing third parties to build extensions and skins and what not. do you think that collaborative markets like pure open source initiatives have kind of served their course or are proprietary going to keep being the way forward or will there be a place for collaborative markets and true open source in the future as well.

27	Rsp2:	So collaborative you mean open source. or what do you mean by it
28	Researcher:	Collaborative in this case would mean were you focus more on building a community of developers like you dont have to govern your third-party developers by strict contractual agreements or just the way it happens in a B2B aspect for example you are having other developers extend your service for instance, which is totally different from when you are putting out there a collaborative product rather than when you have a service that you want to sell like atlassian
29	Rsp2:	So its more dimensional ,but locked versus open source, so i would encourage you to read this report published in 2011 called the open governance index, have you seen that?
30	Researcher:	Yes yes
31	Rsp2:	So that explains open source which is not black and white , actually there is ways you can have pretty locked down projects even if they are open source projects like android. With regard to the kind of agreements you need , its all shades of grey, so you have for example Orange has very clear apis but they are used mostly by tier 1 b2b partners that they work with very closely so that in b2b hackathons they take a month a pre-select who attends the hackathons and spend many days developing developer prototypes, and get their customers in the room and say here are our needs and based on our needs what can you build. so its kind of a closed b2b environment but it works for them . Then we have the opposite which is amazon and AWS which says that any imaginable use case here is the api go and use it,. Open source is a class of governance models about sharing code, and sharing the risk and cost of development. Relatively its the lowest approach to getting external parties to share in the risk and the cost of development. its not so much how you , its almost independent to how you allow others to extend your business. so you might have a browser engine that is open source like webkit but the APIs exposed by chrome may be limited to very few companies. I am just making this up.
32	Researcher:	Yeah
33	Rsp2:	So the browser engine could be open source but the apis exposed by the browser shell might be closed.
34	Researcher:	Yeah ok, Fair enough. alright should we move on to the second theme. We wanted to talk a bit about the considerations when it comes to platform design as well and we found an interesting piece by two harvard students who were talking about the aspect of organizational identity when moving from pipes to platforms basically and what they said was that organizational work very much revolves around building an organizational identity. who are we as an organization, what do we do and how do we do it , and what are our codes of conduct and things like that and our question which is based on literature is do you think that the identity of a company is affected by either applying or transitioning to a platform-based business model.basically how you view yourself, your external relations and how you conduct work. so this will be of course with the comparison of how you treat your core business.
35	Rsp2:	Salesforce is a good example, so dominic has written a case study on salesforce and they say they have over 3 million`developers registered and they say over 70% of salesforce customers are using at least one app from their sales app store. so its a case where a platform played by salesforce is materially influencing majority of their customers.because they are able to do so much more with the platform

36	Researcher:	That is very interesting
37	Rsp2:	We cannot imagine today android or ios today being plat formless, we consider them as one and the same thing. Imagine like having only apple created apps on your phone , there would be hundreds of apps not millions, maybe thousands at best,
38	Researcher:	Yes i guess a good example also would be the fact that when we talk about apple today we think of apps but when you look closely you realize that their core business is still mostly on hardware sales, so i think that is the aspect that we are looking at. sometimes when you transition to platform it sort of maybe give the outside world a totally different view and affects the perception in terms of core business. And also do you think that the company when transitioning to a platform model should remain focused on the core business or is it more important to focus on building a healthy ecosystem. Because there are examples we have read about where companies have completely transitioned focus and ended up losing their core business because they were so focused on the platform in itself and marketing and developing that so they outsourced everything and in the end the whole business model imploded.
39	Rsp2:	I am trying to think of better examples, I am going to run out of battery in 5 minutes by the way. Let me switch off my camera just to save . So a platform like stripe or twilio that cannot exist without the platform. the platform is the product, and there is platforms like mindcraft and the users...well i don't have any other examples actually but i think the message ..the platform itself can be used to send a very strong message about the identity of the brand
40	Researcher:	Maybe on that same aspect do you think that there is any particular strategies used by this companies when they are launching this platforms, how do they reach critical mass for instance in order to attract development. is there specific strategies that come to mind?
41	Rsp2:	So facebook has perfected how they launch new open source projects and gain maximum impact. not maximum but very strong impact from day one because at their conference they were showing charts whereby the overall impact to developers of a new open source project is very much dependent on the impact on day one so you have to get the launch on day one exactly right in order to have a lasting effect and so they are now staging releases of open source frameworks as much like how they would stage a product launch. very well orchestrated from my own nerspective
42	Researcher:	So it aligns very well with atypical product launch. i know because we have had big incumbents like google video for example before the acquisition of google who failed to gain critical mass in the beginning, not enough viewers to attract producers and not enough producers create enough content to attract viewers and the platform just collapsed on itself. SO even this big incumbents seem to sometimes be struggling in their launch strategies basically
43	Rsp2:	I have a 5 more minutes left on my desktop, how many questions do you have.
44	Researcher:	We have maybe we could combine some of them here , we had a couple on boundary resources, maybe one or two and then something about design rules and finally about governance and control one question
45	Rsp2:	Ok, what are the total questions

46	Researcher:	Firstly for platform owners, when it comes to governance and security they have seen that creative freedom and control sometimes are of conflicting nature and the question is like how do companies strategize to balance this, how do they work to assess and evaluate how much to let go to allow developers to be as creative as they want to and how do they maintain basically control of the development
47	Rsp2:	Its more art than science because you have to figure out how you want to evolve the product , you have to understand what people are trying to build on top of it and then whom you are going to give priority to in terms of the things they need to access from your platform versus the things that you don't want to expose , a good example is that web apis on iOS or web browser apis have always lagged...native apis by design because apple wanted developers to be building native apps and not web apps which are portable to android and in the native apis they have much more control, so there is no answer to that ,no single answer to that, every company chooses where it wants to draw the line
48	Researcher:	Ok, Just maybe one more important one i am thinking of, how do they exactly protect their platforms from being infringed by third-party development, are there other ways you could think of apart from agreements and
49	Rsp2:	Well there was a time when apple was asking developers to sign an NDA if you remember, i don't think they do anymore, but in the developer world its more r less expected that if someone gives you an API to use you can use it in any way you like, there are no restrictions, you can depreciate the api , use it for a year and then its not gonna be available but there is no other means of restricting it. you can also curate developers like apple is doing like you have to pay some money to become a developer and apple curate your apps, and like intuit which is a big player also dos the same thing and salesforce does the same thing because they turn their platform very sensitive, its financial data its customer data, they have to very carefully vet or control the applications that go on the platform. But the api level there is not much you can do but at the appstore level you can control what is there and there is much more type control
50	Researcher:	OK, Well maybe just to sum it up is there any other things you feel we may have not mentioned and what is the most important ones you would point out
51	Rsp2:	I think in the course i had spoken about how google controls android using four control points and that is a good example of how like boundary conditions
52	Researcher:	Ok, alright i also have one last question about core business, i mean using third-party development means that you suddenly have to take into consideration your network when developing things or your core product and this change, what do you think it implies to a company who suddenly ..like we are creating mobile phones and suddenly for our own innovation process we want to make our screen a little bigger and that will affect thousands of partners, producers, whatever you would like to call them. Do you think this is a big change that comes with problems and what do you think are the implications
53	Rsp2:	SO the apis you take backward compatibility or in some cases there is forward compatibility. so backward compatibility means that the new apis will be a superset of older ones... and this was symbian in the old days where they changed radically the codebase or binary base which is one of the worst things you can do because you ask developers to rewrite the apps. Like forward compatibility is really important because you giving up an api you promised developers that you can build something with it and like it can last for as long as the platform

54	Researcher:	How do big companies do it then do they announce long before when they are gonna make changes to the core product or
55	Rsp2:	When they take decisions to break forward compatibility its bad news for developers. It's not often that you see it because like twitter closing off apis in a big way two or three years ago. AM running really low
56	Researcher:	Dont worry
57	Rsp2:	I hope i helped you
58	Researcher:	Absolutely thank you so much for your time Andreas
59	Rsp2:	Contact dominic and let me know if there is anything you need with that but he has the business model side covered
60	Researcher:	Perfect .Thank you so much, good luck with your training or maybe you were done with your training
61	Rsp2:	No i did half of it the i have some more later
62	Researcher:	Alright have a good weeknd
63	Rsp2:	Bye everyone

Appendix 4: Interview Transcript [Rsp3]

Rsp3 = Respondent 3

Line	Person	Content
1	Researcher:	Do you have an API that you give out to developers or how to you include third parties?
2	Rsp3:	How the system works is that it's a mobile ecosystem is a platform that we build and send to other parties, so it's basically a software as a service kind of approach. to enrich our ecosystem we have to integrate the platform to other systems and through APIs so our clients have to user our APIs to integrate the platform in the environments.
3	Researcher:	How much time would you say that you have now?
4	Rsp3:	Another 30 min

5	Researcher:	Okay, let's get started. We our looking at the different considerations that companies that aim to use third-party development for their business model needs to adhere to. We categorise these into three themes: developer motivations and marketing, platform design, governance and security. Starting of with developer motivations, there are a lot of them, we have seen that developers are motivated by things like educational tools, support, documentation and of course monetary incentives. We were wondering then in your experience, how do developers prioritize these things and how do you work with that strategically to attract and retain them in your ecosystem?
6	Rsp3:	I would say we go with the first one and the last one, which is monetary incentives and education tools. as a young, upcoming developer you strive to reach a certain point and look for a mentor who you feel is much better than you from a technical perspective who has more experience and is somebody you can actually learn a lot from. Ones you start having that experience, then the other motivating factor becomes monetary, basically how do advance yourself from a monetary and career perspective
7	Researcher:	You have your APIs and then you have these other companies that uses your APIs, they also have their own developers, who have to integrate to your platform. So I was wondering how do you really do your marketing to these companies. do you feel like your market position of the company, your differentiation, we're looking at these clients as third-party development in this aspect. are they motivated by your market position or how do they choose your platform?
8	Rsp3:	It is generally the market position and the profile of the company. because essentially, at the moment, directly is the biggest online payments. What happens is that we built an ecosystem where we offer unified solutions to merchants, both small and big. on the ground floor we're able to collect payments via cards, mobile money, QR-codes so essentially we can collect money based on any channel and from there, one of the biggest pains are reconciliations and settlement and once you solve those two problems, you are able to attract many, many merchants, as you get more merchants you get more clients and bigger clients into the ecosystems. One of the ways we've been able to bring in more clients is essentially via social media, mainstream media from time to time and from our website. We normally get at least 30 daily inquiries, these are leads essentially that you follow and get to close it so those are some of the channels we normally market ourselves in and get more business coming in
9	Researcher:	So it's the addressable market basically that attracts new clients and developers?
10	Rsp3:	Yes
11	Researcher:	You work in a b2b context so when we talk about things like system lock-in, making sure that developers don't move to competitors it's contractual agreements i supposed, do you do any other things like offer very big SDKs, how do you raise switching cost?
12	Rsp3:	Define switching cost?
13	Researcher:	Basically whatever means you use to make sure that they don't use competitors platforms instead

14	Rsp3:	Okay, it's pretty simple because a client go to a competitor simply because your service is not working as it's supposed to be so essentially your competitor is having an edge over you. So what we try do to is to always innovate around the solution that we have and as well, getting to know what your competitor has on the ground, knowing what they are working on and trends within the market and then also we have the strategic partnerships with AMEX, visa and mastercard so any new innovation they come up with you get to be the first to use, you know the first to roll it out to new markets. that is one thing that we are able to do essentially that helps to retain our clients and the second one is basically account management and understanding you clients, personalizing your client's solutions okay so once you understand your client you're able to basically give them a solution that will intrinsically solve their problem, as long as you solve your clients problems, you are able to retain them
15	Researcher:	So it's more like diversifying your value proposition?
16	Rsp3:	Yes, exactly
17	Researcher:	Should we move on to platform design? how old is the company?
18	Rsp3:	13 years
19	Researcher:	Okay, so you're fairly old and I presume an incumbent in the industry. a lot of platform companies, when they either switch from a platform or starting up for the first time, however you choose to launch your platform weather it's b2c or b2b contexts have to struggle with what we call critical mass, which basically getting enough users and participants to make the ecosystems truly fruitful
20	Rsp3:	Yes
21	Researcher:	We are wondering how your company and other companies strategize this, how to get enough both producers and consumers?
22	Rsp3:	When I look at it in hindsight, because I've been in this company now for one year and two months, but before that I was working for a company called ... that's actually where I met Sylvia, I look at both companies from a strategy perspective and I think initially once you get your strategies right you understand the kind of solution that you're going to be building so I'll give you an example, my former company got started by wanting to cell ringtones, basically the tune that plays when someone waiting in hold for a call. The strategy was to basically to become a billion dollar business by getting a hundred million users to come onto the platform transacting at least one dollar per customer per month. what that meant to us was that you couldn't build a pipe based kind of business for that kind of an approach. they needed to get access to the musicians and as well get access to a value store which is the banks and mobile money solutions to get the end customer. We had to think beyond what was the obvious components of the ecosystem, in order to provide the end user everything he needed in one place. for a customer to buy a ringtone, they need to have access to their money so what they did is they built a mobile VAC... and that was to access the customer's money and then got access to the customers mobile bank by integrating to the mobile network operators and then how to deliver this service by integrating it into the app
23	Researcher:	So you can say the enabled all aspects of the ecosystem

24	Rsp3:	Yeah so know when you come directly online, it started because the co founders had an airline background, they had worked in the airline industry for a while. so they started by doing a booking system for one of the airlines and then they notice that most guys that most guys that are traveling by airlines will require to have bookings, that means that they need to access hotels and then as well, they built a system where they would enable somebody who's coming probably europe, coming from a foreign country, they can pay via card for their fly ticket as well as make their booking and then also do their shopping via card so they built a system that had a module that could accept payments for all kind of cards and that means integrating the schemes which is visa, amex and mastercard. So, essentially, the platform became bigger and bigger and bigger. Now suddenly interesting clients such as booking.com, expedia and then from that you just basically cloud the system. I don't know if that answers your question but..
25	Researcher:	What I understood there is identifying a need that was not initially there so the strategy was to provide a service that aggregates
26	Rsp3:	You need to get it right the first time because it is expensive for a company to redefine itself.
27	Researcher:	Do you think platform companies have to be able to sustain losses in the beginning
28	Rsp3:	Yes, especially with mobile business, you have to move quickly and you have to be the first in the market with a certain solution to get the critical mass. before your competitor realize that they need to do the same thing. at some point, you are required to get the right talent you need to put in money to the technology that you're putting in. So in the first few phases of the business you might have to sustain losses. where I was in my previous company, you find spikes of buying that your suppliers have built up to maybe once every three months and then you work on a project you need to find the project team which means that for 90 days in between the spikes you're not getting paid. Because there are these spikes, most companies go and look for investment to make sure that the spikes become constant. it becomes one level operating approach
29	Researcher:	Okay, nice. I think I just want to ask another thing on a different angle. I was wondering, you have SDKs and APIs, we call these boundary resources, the way you interact with the third parties. Are there any other resources you have and how to you prevent them from infringing on your product for an example?
30	Rsp3:	Let me see what I understood what you mean with boundary resources. You are interpreting these as channels that basically customers can use to access your service, right?
31	Researcher:	Yes, interface yeah

32	Rsp3:	<p>The main platform, we have what we call an integration layer. that is what is exposed to our customers. and within these APIs, it doesn't really matter type of channel they use, use the mobile application or they use an SDK, all these platform integrate into one API that then in turn propels them to the main solution in the ecosystem. For an example, depending on the type of business that you run, a merchant may say that I want my customers to pay by directly online, I would not give them an application to that. I basically just expose them to one of our APIs to let them integrate it to their website so that is one. the second channel is that small merchants who basically have walk-in customers. let's take an example of a hotel where someone just walk in and make an accommodation booking. we can give that person the application, we have that merchant spacing application that allows them to do sales transactions, then as well we have the end consumer who we'll expose to one of the applications that they are actually able to consume our services from. An example like Juvia or Uber, instead of them using our application, we give them an SDK so they use that to tap into the platform and then just basically transfer the application for end customer to use. it's basically maintaining the main platform to be as easy to access via various channels and making it standard you're able to harmonize your product across the various channels.</p>
33	Researcher:	<p>Great, how do you prevent these externals guys... what rules do you set for these people when they integrate with your product? how do you manage these relationships?</p>
34	Rsp3:	<p>From a technology perspective?</p>
35	Researcher:	<p>Both</p>
36	Rsp3:	<p>Okay, so the nature of our business, especially in the card industry, there's a lot of fraud that happens, the first process is what we call a KYC process - know your customer process. ones they tap into the platform they have to accept an agreement and we request for their relevant documents like financial statements etc. and then we hand it over to the fraud and risk department so, they will go through the documents really thoroughly for the all the alleged to be merchants. once the vetting has been done they will approve, mr sales rep you are good to go. so now the integration process can begin depending on the model we want to use. For an example, if it is a big merchant maybe Uber, depending on the design of our API, there are several models that we can use to integrate with their API, since we are tube-based kind of a system when we receive a transaction from a platform, we get a request and then we send the status to the merchant and there are two ways; the merchant can either use our API to handle transactions, or from our side we immediately realize transaction notifications with the merchants. So looking at those kind of setups there a number of things we normally look at. we look at security, load times and turnaround times, when you put some of those things into perspective we are able to basically implement a solution that caters for all those kind of things. For a big merchant, things like loadtime come into play, security of course applies for client of every size for a certain scenario, instead of the merchant using our APIs we actually do it the other way around, we send their APIs, we send them as status to their platform, in that way we avoid the merchant querying our API even if there is no transaction behind to preventing the load on our site and then as well on their site they are able to see what more resources that are on their platform. these are some of the matrices that we normally use and some time you need to do monitoring and see "is this guy using the correct algorithm to query our system, is there a queue that has ot been processed and then why has it not been processed, what application is not running, basically the monitoring tools as well</p>
37	Researcher:	<p>Just in the interest of time I want to ask something in the next three minutes. One thing is, how important is your pricing model for instances, do you have a constant pricing model for these third parties or what do you really offer them to maintain your power and trust and another thing is, how do you balance control in terms of what you expose and what you don't, do you find that you have an interesting balancing act in terms of what you should</p>

		give to third parties or not, are there any challenges there?
38	Rsp3:	From the pricing model we normally have revenue share kind of an approach, we use what you call a tap model. we have a standard charge for merchants who have low volumes but we give them an incentive if they grow their transaction which could maybe be 2 percent of the value of your transactions, that is how we brand them for an example if there's a hundred thousand transactions per month, this is the percentage that we will charge you, if it's between hundred to five hundred, you'll reduce the percentage of cost to a certain level so at a certain deal it level out to hyper volumes. when it comes to control, one of our competitive edges is our fraud and risk department, we don't expose those practices too much to our merchants because our competitors don't have a lot of efficient systems that handle fraud. for an example, maybe your card has been stolen and it's being used by some nigerian bandits then we will take care of that for you and that's one of our competitive edges and we try to limit the kind of information on the intrinsically functioning models of the platform, besides that I think that the rest is normally open
39	Researcher:	I have a final question that addresses all your experiences as developer or working with platforms in general and I was wondering, when we present these three themes: platform design, developer motivations and governance and control. Imagine you would starting of your own platform business, whatever it would be, do you feel like there is something additional that you would take into consideration besides the themes I mentioned
40	Rsp3:	Okay think from my experience, one most important things is getting the solution right, looking at all the aspects of the solution and knowing exactly where you're going to be, where the business is going to be and based on that, building a light inexpensive platform/solution that can handle a lot of transactions at a go, that builds in the load perspective into play. and something else that generally comes into play in tech firms locating and hiring the right talent within the business to handle such things so you find that in a starting point for certain businesses where they have to look for a certain kind of talent, which becomes very expensive so if that kind of talent lacks, they may have to hire a lot of people and then start doing the learning process. so for me a lean and efficient team is very important. that controls the expenditure of the business and becomes more worrying as the business grows and as the volumes grow, that really minimizes the losses but then as well, going open source is another thing that I would recommend because it kind of reinvents what is already out there, someone else have solved a problem within the market and you can build on that to make your product more efficient as well
41	Researcher:	So I presume that one important aspect is remaining focus on the core product and not lose yourself in all the different aspects of the product
42	Rsp3:	Exactly
43	Researcher:	That's all from us. thank you so much
44	Rsp3:	Thanks guys
45	Researcher:	Bye
46	Rsp3:	Bye

Appendix 5: Interview Transcript [Rsp4]

Rsp4 = Respondent 4

Line	Person	Content
1	Researcher:	Maybe just in a nutshell, what would you say is your responsibility as a system architect
2	Rsp4:	I think it entails, its two things, so first of all its to ensure the current business domain model is not broken, and secondly is to continuously find out ways in which we can be able to improve on the domain model and make it work for changing times
3	Researcher:	I think that quite great coz in this thesis we are looking at how businesses could literally extend their business models using third-party developers and i thought Jumo would be a good example because i know you are working on an open API so basically you have the aspect of exposing your services to outside innovation
4	Rsp4:	Yes
5	Researcher:	Maybe just to dive in directly, what would you say are the main developer motivations like in the aspect of b2b and trying to integrate with other companies for instance what would you say are main developer motivations . there could be monetary incentives, support, education, documentation, what would you say are the main developer motivations for instance
6	Rsp4:	So you mean why would a developer participate in it or why would a business participate in it
7	Researcher:	Yeah so the thing is , we think are going with the concept that even if you expose and integrate to another business, that they still have developers on that other business, so it is b2b but one way or another you are still going to target a developer from that other company. but even so generally what would you say are developer motivations as a developer when you want to develop for a platform for instance
8	Rsp4:	I think there are several things you would look at and say one of them is the experience you get by developing for third-party, for example if you go like lets say , if i was an external developer and there was Jumo API. By me developing for example against that particular api it gives me a bit of experience on how for example Jumo works and in future if i wanted to work for Jumo and i was like , by the way i have actually done an implementation of your API it gives you a bit of extra points as far as for example the interview process is concerned but also as it being part of your CV and in the age where most businesses are going towards b2b integrations because businesses have figured that at no given point will you be able to for example be sustainable if you are not continuously integrating with other businesses, so the fact that you can actually be able to implement some of these interfaces gives you an added advantage in the job market . i am not sure there is any direct monetary benefits or any monetary motivations because Jumo is not going to pay you simply because you have implemented their API because you are not necessarily the target market for them so ..yeah
9	Researcher:	And do you think that the market position of the company for example would be something one considers?

10	Rsp4:	Yeah i think that it is something that would be considered because for example , lets compare Jumo with a company like Google for example .if you put lets say there is a jumo api and a google api, most likely most developers are going to develop for google api because they are well known , you know they are known to follow interesting and international standards kind of a thing, so at any given point the muscle of the bigger company is going to win over the rest of the other characters.
11	Researcher:	And do as jumo , how would you suggest that you evangelize your product to like the developer community for instance.
12	Rsp4:	I think by participating in things, for example tech workshops, at least at those workshops you are able to showcase some of the things you are doing and also evangelize and give examples using your own APIs and then i think we might maybe in future for example once the api is public ready you know, sponsor things like hackathorns to build against that particular api , encourage other third-party applications to essentially use our api for example for payment integrations, that kind of thing
13	Researcher:	I totally agree, and do you think there is anything specific that you do to lock in the developers, like now you have evangelized and you have now have your developers in the platform for instance like how do you increase your switching costs to just retain them and not make them go to someone else implementing the same thing for instance
14	Rsp4:	I think at the end of the day it all depends on the potential customers that for example third-party developers will see in your platform, so as long as you ensure your platform has enough number of partners and customers, naturally developers ill go towards places where they have bigger impact, they can be able to develop applications that have a wider reach in terms of the customer base that you have. so for a business that exposes public apis keeping the total number or having a high number of customers is always going to be a big big sticking power
15	Researcher:	I agree actually, and well, when you think of , we have collaborative markets , let's say for example linux they offer, they are not really profit making, they operate as an open source platform but then again we have this other proprietary platforms like lets say for example Jumo. Do you feel like there is more , people are moving towards proprietary and this collaborative markets are kind of have run their course already?
16	Rsp4:	I fell what usually happens is, it all depends on the philosophy a particular organization has. so there are disadvantages and advantages of going either direction, for example if you go for open source, as much as it is public scrutiny , it also has its own issues, for example the fact that i know which technology you use internally, i can go out there and research for its vulnerabilities and then use that against you , on the other hand if you are using off the shelf software it means it is not exposed to the public so most likely there are things it suffers from that you may not be able to know because you also dont have like a widespread use of that particular technology. so yeah everything has its own pros and cons
17	Researcher:	Yeah i agree, ok so i forgot to tell you earlier that we have like 3 themes so the first one was developer motivations and then platform design and governance and control. so am done with the developer motivations, i dont know if there is anything else you feel in terms of what motivates developers that you feel you could mention?
18	Rsp4:	Yeah i think its pretty, its quite comprehensive i think

19	Researcher:	OK , so i think i will go on to like , platform design. so when you think about an organization like sometimes you have let's say Jumo for instance you have a specific identity and do you feel like having to apply this platform-based business model having to expose your api do you feel like, it almost corrupts your identity or how do you feel like it has any changes to you do as jumo
20	Rsp4:	I think on the contrary, i think it enhances your belief. some of the motivations behind people doing the kind of things people do is because, i mean if you are not motivated by money you are motivated by what i would call sudo-ego where if you are doing some cool stuff you want the world to know about them and that is pretty good for you, so yeah that is how i would answer that question
21	Researcher:	So its like generating network effects in a way
22	Rsp4:	Exactly, yeah
23	Researcher:	SO i guess your suggestion that am getting from that is that the company should always remain on their core product and not like, switch focus to adapt to the ecosystem or is it that they just use outside people to enhance their core business
24	Rsp4:	Yes , if you don't want your company to develop tunnel vision , its important to have this external people coming in and telling you by the way this is for example , let me use , lets say i wanted to come up with an open source software, there is me, one person thinking about how it think my design works, but then the fact that i open it up to other people, i also get professional and peer critiquing which essentially improves the overall product
25	Researcher:	yes yes i agree. SO clearly before you had mentioned how developers would want, you had mentioned the aspect of having critical mass like if you have a certain number of users it kind or attracts more people, like the more number of customers or users you have on your platform then it will attract more developers as well. so do you think there is any specific strategy to attain critical mass
26	Rsp4:	It's purely a business call, not a software or engineering point of view. let me give you an example. Android, its a very, ...android was not the first open source mobile OS there were others before it. but what google did, they said to the manufactures we are going to give you this thing for free , no strings attached.. have it ,install it in your devices and that essentially sparked developers to develop for android because suddenly every device that was coming to the market had android installed . so using that example, for jumo, lets be speculative and say Jumo develops or rather says, hey our wallet is now open source, you can actually develop for it and its going to come with every new android device that comes , having this interesting capabilities for money transfer and all those kind of things. that means for every new person who buys this new phone, they potentially become a customer for the developers right, so that's exactly what i meant
27	Researcher:	So i think maybe during this initial launch the company will need to have a way to sustain some initial loses before having a way to get some real profit out of it

28	Rsp4:	Yeah but most of the time if the ecosystem is proper for example, for google it was just ensuring they are able to have all their applications developed and you can easily access their application, for them that was their motivation right, so it would be the same thing as saying what would be the total cost of us trying to develop for very many different types of operating systems and then try and sell that to the various end customers and then figuring out that its way cheaper if you actually sponsor an operating system , give it for free to the manufacturers and the manufacturers are going to actually do the rest of the job for you, as they will just have to say this hardware is running the best software and lets run it, but then as a rider you get a benefit that this operating systems come with the google application for example so its a win win situation
29	Researcher:	Nice , i agree, makes a lot of sense, so if we talk about boundary resources meaning adks and api, that is ideally how you connect to third-party developers, i was wondering is there any other boundary resources you use to engage this boundary resources you use to engage this developers
30	Rsp4:	Yeah all of a sudden this becomes your major marketing channel so you have to sell it as a product . if you are releasing an app into the market, almost the same about of effort would be required only that selling apis is a bit different because its a product that creates a product so its a bit different and you have to make it like, ooh this this is easy to use, or it can do this, this and this
31	Researcher:	So you have to sell its capabilities right
32	Rsp4:	Yes
33	Researcher:	So during the design of this apis are there any specific considerations you put to ensure people do not infringe on your product for instance
34	Rsp4:	I think when you go open source there are certain things you have to be aware about, that sometimes you are not necessarily selling the intellectual property to the backend product remember that, so this is just an interface to interact with backend product so i don't see any issue, literally by my to use, for example the Mpesa API, does not tell me exactly how mpesa works in the backend so its protective in some way, so it's not abad thing to show people what you are doing and the fact that you have more developers aware about that technology it gives your technology an upper hand and you become a discussion when people are trying to make decisions about which technology to use. the fact that your api is out there gives you an upper hand against a technology whose api is not there
35	Researcher:	Interesting. and i was just wondering the way we have systems and user interfaces, is design and user experience a thing you consider in api design
36	Rsp4:	Yeah i think at some point i said you need to look at api as a product and if you do that then all this considerations become very relevant in terms of how you package which means that it has to have a nice UI, the developers are able to interact with it, you have a good support mechanism and yeah so its of importance
37	Researcher:	Nice. just the last question on this category i was wondering , and this may sound a bit repetitive but is there any other rules you give them or its just agreements or is there any other rules you have for people using your api
38	Rsp4:	There are never rules about how you implement an api as long as you dont use that api for illegal purposes, its essentially upto you to go out crazy with it and actually implement an

		dbe creative
39	Researcher:	Maybe just to go into the last one on governance and control. i was wondering like ,you have this ecosystem of developers and what not on your platform, would there be any like, i mean how do you keep them engaged for instance in some sort of lean communication the way linux had this community , how do you really communicate to them
40	Rsp4:	Most open api specifications or implementations have what they call mailing lists, you have community forums where people can actually ask questions , and get help from the rest of the community, so there exists such kind of communication channel to use
41	Researcher:	Interesting and do you have some sort of pricing models
42	Rsp4:	No its free
43	Researcher:	When we talk about just exposing your api we realize there is usually some sort of conflict when you have to facilitate creative freedom for third parties but at the same time you want to exercise control over your platform, have you experienced such conflict
44	Rsp4:	I think it's not necessarily conflicts but its internally being aware that for example you might have thought when you design an api that this is wat it's going to be used for , but then the moment you expose it to the community they take it and make a different product out of it so sometimes being able to wrap your mind around and accepting that fact , for example let me use a kenyan example of mPesa, initially it was supposed to be a small group saving thing as the initial purpose for it but then , not necessarily through open sourcing it but it ended up becoming a money transfer solution, so you will find situations where you give something out as an api and you are expecting people to build ferraris with it and they end up building Lamborghinis and other own custom made cars, i mean that is something that every business should come to terms with . lets say ok we are not going to restrict what you can do with this thing but just treat it as a product that builds other products. How the implementation happens, you leave it to the prerogative of the developer
45	Researcher:	ooh yes, i just forgot one question about for example when you have to make changes to your product for instance or API how do you communicate this or does it affect the ecosystem in any way
46	Rsp4:	You always make the apis backward compatible and always versioning
47	Researcher:	Totally agree with that . well i am done on my part so we have talked about governance, design, and having to set design rules with the third parties and talked about consideration of third-party developers, is there any other thing you feel would be important to mention especially at the point where you have to launch a product to the market
48	Rsp4:	mm no i think the interview is pretty comprehensive and i think it covers most of the pain points and some of the things that essentially most people dont have as clear so i feel its quite comprehensive
49	Researcher:	ok, thanks alot i think that it from my side and incase i had a specific question i forgot i may maybe follow up if that is ok with you
50	Rsp4:	No problem i am ok with that

51	Researcher:	Thank you so much and have a good day Ken
52	Rsp4:	Alright then have a good one
53	Researcher:	Bye

Appendix 6: Interview Transcript [Rsp5]

Rsp5 = Respondent 5

Line	Person	Content
1	Researcher:	Firstly, can we record this interview
2	Rsp5:	Its ok i dont think its a problem
3	Researcher:	Ok cool, yes so well maybe you can tell us a little bit or maybe just a summary of what you do like in your job description for instance
4	Rsp5:	So i am a technology solutions manager for cellulant, basically my job entails designing solutions . i just have a look at the customer requirements and come up with the design of the end solution and engage the team of developers to have the solution implemented , tested and deployed on live. and the last this is supporting all this systems that we take on live, so we normally have service improvements and also making sure that the TAT and the success rate is good . we have an sla of what should be the tat and what should be the success rate for this transactions
5	Researcher:	Earlier you has sent me something about the Mula platform, could you maybe summarize for bjorn maybe what the purpose of it is
6	Rsp5:	OK i think i will start from wat cellulant does and go into details of what mula is all about
7	Researcher:	Sounds good
8	Rsp5:	Cellulant is a payments system, basically we develop payment solutions for our customers and also for a b2b businesses, this is for banks and merchants like gotv, dstv, where we make sure that we provide a wallet or a means to pay for your services and the means to get subscribed to the service. so for instance we give a customer the option to buy electricity tokens and the customer has the ability to pay via all the mobile money that we have within kenya lets say in kenya there is mpesa, there is airtel money and a new one called mcash, so we give them the ability to buy electricity tokens using the mobile money platforms and also at the same time they are able to pay via their banking system , we have like a mobile banking platform and customers will be able to pay via any of the banks that they have accounts with, and once the customer has done the payment then he will be able to integrate to the merchants or service providers to get the end product which is for instance electricity, gotv and other services like start times
9	Researcher:	So how does mula work?

10	Rsp5:	The mula platform is divided into two , there is the checkout platform which is what we give to third-party companies or individuals that want to integrate to our system, and third-party they will be able to charge the customer and fulfil the request of the customer. what we normally do is we give you the platform and you give us your customers then we do a share basically which could be 50/50 or 60/40 kind of sharing in terms of the commission and that is for the check out, but for the mula app that is a core product for cellulant where we provide this services directly to the customers
11	Researcher:	Interesting. SO i presume you have strategized means to attract and retain this developers, getting this , you know, however you recruit clients almost to use your APIa and to integrate with your platform and system then you have to know a little bit about their motivations i presume like what do you offer this developers in terms of educational tools, support, documentation, maybe monetary incentives as well
12	Rsp5:	Ok so we have provided a couple of services to the developers who want to consume our api and the first thing we do is basically documentation, so we give them a documentation of how and what are the available services coz within our system we are able to do a query bill, that is basically provide the , lets say for instance its electricity we provide the meter number of the customer and be able to query the bill for that customer or basically check how muchlets say like post paid where you have paid electricity after usage so your able to check what is the bill of the customer before the customer even makes the payment , then what we also provide is the post payment service where you are able to push a request to charge the customer and fulfill the request and give you an option where you are able to query the status of the payment because the way the system is structured is asynchronous so you push a payment you get an acknowledgement and then be able to query the status of the payment once it is fulfilled
13	Researcher:	And now what would you say is the major motivation of this third parties to integrate with mula or cellulant for instance.
14	Rsp5:	Ok , one of the greatest advantage for third parties is that you get to all the markets we have footprints in , for instance we are in 11 markets in africa and we are connected to SMS USSD , and this is to all the MNOs we are connected to them via SMS , USSD, Mobile Money and airtime, which means that if you connect to cellulant you will do one integration that basically maps you to 11 countries, so this is close to 3 MNOS per market which is equivalent to around 33 MNOs and also connected to around 80 banks and this are some of the major banks like ecobank which is a , almost in 33 markets currently
15	Researcher:	Are you the biggest actor in Africa for this service
16	Rsp5:	Yes i would say we are the biggest actors coz we also , i think we had Google and Opera mini come through us coz of the coverage advantage that means that they have one integration then they have access to all this markets
17	Researcher:	Then that gives a good market position to be able to attract this users right?
18	Rsp5:	Yes yes
19	Researcher:	I was just wondering how do you really evangelize your platform to this third parties like are there any specific ways in which you do this?

20	Rsp5:	Ok, currently we haven't done alot of marketing on the checkout coz most of the marketing has been done on the mula app coz they are trying to push this as a product coz it goes directly to the consumers themselves but according to what the strategy that has been laid down is that in the end we will be able to push this through media and what we are going to do is that the commission share is going to be either 40/60 or 30/70 and this is to the advantage of the guy connecting to our system, meaning that they connect to our system they make more money
21	Researcher:	So it's a monetary incentive in a sense
22	Rsp5:	Yes
23	Researcher:	So when you have gotten developers into your ecosystem and when they chose to integrate with your platform, what do you do to make sure they dont live for a competitor, how do you lock them into your system. or is that strategy just in any way
24	Rsp5:	We make sure that the system is fast, reliable and gives you the accurate end results, most of the third-party developers need to prefund their accounts , but when we are doing the calculation of how much you have prefunded and how much you have consumed then it needs to be transparent. At the same time we give them an interface where they are able to login and filter out the reports basically they are able to check how many transactions they have sent through , how they are processed and all that, the other thing we make sure we provide is tat and success rate , coz as long as.
25	Researcher:	What is TAT?
26	Rsp5:	Turn around time
27	Researcher:	Ok
28	Rsp5:	That means that as a consumer i wouldn't want to buy airtime and wait for 30 minutes for that airtime to come through, it should be a matter of seconds. so we make sure that the TAT is low and alos make sure that the success rate is very high meaning that if i send 100 transactions , they should be successful and we have less failure rates. Then what we also provide them is that we are going to more markets, i think we will be going to egypt, congo, rwanda and another fourth market in west africa, which means the coverage will keep growing and in the end we just want to be in the whole of africa
29	Researcher:	That makes a lot of sense actually. Ok so i don't know, maybe just to backup a little bit we have seen alot of , you know just from a high level perspective you know how lets say linux for example opens , collaborative markets vesus cellulant which has a proprietary aspect whereby you offer a service but is sort of closed compared to the bness model of linux . would you say that many people are going towards the proprietary side or just having an open api or open source model is still viable.

30	Rsp5:	I think, I don't know how to look at this because you can still have open sources and still make business for yourself or the company and as an example you look at what we have for safaricom, before safaricom had closed their services and api to a couple of service providers and for you to be able to access these services, you had to go through the service providers and that means that your cut becomes smaller because your share is between safaricom, your service provider and yourself. but recently around last year march they opened their systems to anyone who wants to connect, so their apis are out there, if you want to connect to their system for you to do mpesa transaction then you can do that. so you are able to make money still, you just need to design your system in such a way that in the end you look like a service provider although your system is open source and anyone can use them and have access to them at any time
31	Researcher:	Ok thank you, sorry sylvia did you have anything more on motivation
32	Researcher	Maybe just on a personal level, what would you say that, just to summarize this topic of developer motivations what would you say is the most striking thing that appeals to a developer in choosing one platform over another
33	Researcher:	Yeah speak on your own experience beyond where you work now as well of course
34	Rsp5:	I think when integrating to a system what i would look for is number one in terms of business i need to have the addressable customer base, if i connect to a system then, let's say there are two platforms, if i connect to platform A which has 20000 customers and i connect to platform B that gives me around 1 million customers then i will go to platform B coz i will have a larger addressable base of the customers. Then the second thing would be the ease of integration, i think there are some systems especially the legacy systems it becomes very hard to integrate with those systems coz they have so many instructions and commands on how to integrate with the system compared to maybe a restful api that adjusts, its easier to integrate with that system. then the other thing i would look at is support. how well do they do their support, so i log a ticket and wait for several days for it to be responded to. i think something else i would look at is how well their systems are, are they reliable, success rate, TAT, SLA, in terms of just the performance of the system, so to me those are the 4 major things i would look at.
35	Researcher:	Alright, cool. That was very clear. Alright should we move on to questions about platform design, same thing here, taking into consideration like all the places you have worked and your experience as a developer basically and or a platform manager or however you know, all the experience you have because it is really interesting for us of course to make generalizable conclusions, so speaking of platform design, when pursuing a third-party development business model, do you think that a company should remain focused on the core business and the core process or is it more important to focus on creating a healthy developer ecosystem. How do you think companies should prioritize for example if you are moving from a pipe based business to a platform business like in the case of amazon or something like that

36	Rsp5:	Ok , i have two options the way i look at it is that i can either have a separate company or a third-party do the integration for me or do the platform for the system for me or i can have a tema within the company do that and the reason for this is that lets say there is a situation where i have a third-party do the integration for me, and the example i will give for this is like what we have here in nairobi water as a company that provides water for the residents within nairobi, so their core business is basically providing the water and making sure evry home has water but at the same time they need to collect the bills for this customers. and they need to facilitate a way this customers are able to pay their money digitally whether via banks, mobile money or able to go and deposit money in the cashier or something like that . so for something like nairobi water i would suggest if they have a separate third-party service that provides the platform to make payments and....let me just charge my laptop, just a minute.....ok i think i am good
37	Researcher:	Ok
38	Rsp5:	As i was saying for nairobi water, their core business is providing water and that in my opinion is what they should focus on coz that is the mission and vision for that company, providing water, and that is what they are good in,. the bit where they need to collect money from the customers, if they give that to a third-party service that is good at doing integrations to mobile money and bank services and providing the expertise then to them it will be an advantage because they dont need to think about how the platform works and all that they just need to know we have provided water to end users and we have been able to collect money from them without the need to go into details of how does the system collect money and all that
39	Researcher:	Ok , fair enough, good answer
40	Rsp5:	For companies that don't deal with it staff then i think that will be the best model for them. they focus on the core business and outsource the it part to another company that is good at doing that
41	Researcher:	And do you feel like when you are not actually outsourcing but you are just extending your business model by providing an api as an it company , do you feel that has the same results or does it sort of corrupt your core business
42	Rsp5:	Just repeat abit
43	Researcher:	I was just clarifying that if the same example would apply for an IT company that is not outsourcing but just extending the business model like the way cellulant for example is doing,, so what i get is you still feel that they should stick to the core business model
44	Rsp5:	Yeah i think for an IT company i think they should have the developer or IT part within the system, `for instance what we have as amazon or what we have as Jumia. cos their work is basically digitizing sales if they have this within the company as an IT department to handle the integration to the api, handles the it stuff, that means that they would be able to make better products coz then the ideas would just trickle from the business part and come to IT and It guys will come up with a solution for that and able to integrate , and come live with the new changes and all that .
45	Researcher:	Ok that makes a lot of sense, and maybe on that same aspect i was wondering like when you have let's say you have developed that one product lets say like mla api and you want to launch this, coz we realise sometimes like google videos failed because they were not able to gain what we call a critical mass as a launching strategy so how would you advice

		in this case in order to gain market share at launch
46	Rsp5:	I think that is very important that you have brought about , because one of the suggestions we had about mula was can we outsource the marketing bit, can we give it to someone who is good at marketing so let them think about the billboards and how they get into the media , we just have someone who is an expert in that field and give it to them to do the marketing. coz having a whole department for marketing proved to be costly and currently we only have 4 personnel for marketing , meaning we are really under resourced for this . if we could have this outsources to a company that have a whole workforce to take the product out to the market i think we would have made a better leap for it.
47	Researcher:	So then it feels like you have to be willing to invest alot at the beginning and maybe be prepared to sustain some loses as well
48	Rsp5:	Yes yes
49	Researcher:	Yes that is usually a very hard face but maybe to just get a little technical on boundary resources on how you interact with third parties so basically the main things you have identified is have apis and SDKs , is there any specific things that you take into consideration in the design of this APIs and SDKs and how you distribute them
50	Rsp5:	What we normally make sure is that accessibility of the system that the users even if you provide them with the documentation of the system then you should be able to provide them with a staging environment where the integrators will be able to connect to the system and confirm that they are able to consume the services within what you have provided as the api. then once we are able to do connection to the apis and you are able to test and confirm that the services is ok, before you migrate the services to live or basically have you consume the services on live we normally make sure that we sign a contract with the integrators or third-party developers , this is to make sure that we formalize what happens within the system
51	Researcher:	How do you prevent developers from infringing on your product?
52	Rsp5:	Yes, that is to make sure that they don't misuse the service cause you could have someone that is connected to the service that keeps posting payments that end up jeopardizing the performance of the service. Contract binds them to make sure that we agree on how they use the service and those include our expectations and what they can expect of the product
53	Researcher:	So it's contractual agreements, API, SDKs, that is what you use to make sure that this doesn't happen or?
54	Rsp5:	Yes cause every user has set of credentials so we're able to monitor their activity
55	Researcher:	Okay, well, maybe just to look on the design rules, when you launch a platform that extend the business model, I presume it is also important to take into consideration your competitors and how do you strategize for competitive advantage?

56	Rsp5:	What you need to have prepared is a very firstly a working product, a product that provides the best UX to the end user, that means that the system is always, that it's easy to use which means I can buy airtime in one of two steps, I don't need to three or four steps to get what I want. we have a system called water planting, you buy it for your SSD, you enter the amount and then hash which means that you are buying this amount of airtime for this number in just one action, it gives airtime to that number, either your number or someone else's number. those are the sum of the things that we normally do you know, make sure that we have the best user experience for the best user
57	Researcher:	I think you should lend that to airtime, it was the same when I was buying airtime by this SSD session and the process was way too long
58	Rsp5:	What we noted is that as long as you have a very strong product and a product that has as many services as possible, that means that if I have moula on my system, than I don't need anything else, I am able to pay my rent, I am able buy airtime, I'm able to buy my electricity, I'm able to pay for my TV-time, start subscriptions and also able to pay for any other services you know. I can also make payments to other individuals by transferring from my bank account to another bank account
59	Researcher:	What happens when you have to change your system, since you have other external parties it means that everytime that you make a small change the same has to be propagated through the ecosystem. not to ruin the work of all the API integrators and third-party developers, so how do you work with that practically to make sure to you keep everybody happy? how does having third-party development affect your own development processes, for the core business?
60	Rsp5:	What we normally do is to make sure that we have as minimal changes to the core system as possible we are the ones who come up with the core system, what we expect the developers to integrate to. we enable the third parties to do post transactions only
61	Researcher:	When you're making changes, do you announce that far in advance to these third-party developers?

62	Rsp5:	<p>For sure we have very minimal changes end up affecting external customers but the reason why is that we make sure that regardless of what we do on our end, the developers never need to change how they integrate to our system. we basically send out emails and make calls to all our customer on any changes that we are making internally. Let's say that there's a network change or we're moving from a datacenter to another datacenter, we always communicate these changes. We need to make sure that these changes doesn't in any way affect any customer's interaction with our system. The only change that affects them is that if we're moving to a new datacenter, what we normally do is schedule for maintenance. we communicate that "next week we're gonna do a maintenance on our system, we are moving from datacenter A to datacenter B and we need you guys for around 2 hours and we normally do this during offline hours, you know when we don't have guys interacting with the system most frequently. We then ask them to be available for tests and once we do a changeover that takes normally two to three hours, they are able to test the change within the system to make sure that data that is fueling the system hasn't changed and to ensure that all customers can access all services on the new system, that's when we schedule for the actual change. The actual change happens when we can ensure that everything is actually working the way it's supposed to and that everyone can interact with the new system that's when we schedule for actual migration which is the same exercise we do for maintenance. we use a testing system and usually about 90% of customers are able to interact perfectly with the system the first time and we're left with the last 10 percent who we support as much as possible to fix the issues that are popping up. what I like to mention, it has a lot to do with how you design the system, if you design the system in such a way that any changes within the should not affect how the customers interact with the system, that is well-thought design of what you want the core system to be and normally you find that we have other customers request for something extra, they want different types of interactions with the system so for those customers we normally have wrapper scripts, what they mean is a scripts that wraps the special request of the client. it's a different system, either it's a legacy system or it's some other request and then that wrapper script will basically transform that request to one that is working with what our core system expects. that means that we can integrate to any customer with different system or different ways of pushing their requests even though they don't want to change their own systems and the biggest example is what we have with Banks and Airlines, banks they have a core banking system that for instance 224 or any other core banking system, it doesn't change how they interact with our system because you find that 224 does ISO requests but our system expects rest-forum requests so what the wrapper script does is the receive the ISO request and transforms it to a reform request before it's sent in to our system. that means that we normally don't change our coresystem, the only thing we change is the interface that interacts with the customer.</p>
63	Researcher:	<p>That is a very interesting approach and I'm just wondering maybe, is there any other design consideration that you feel that we have left out? To maintain power and trust to your developers with your third parties, are there any other considerations that stands out?</p>

64	Rsp5:	Well, we're also using an SLDC from discovery, design, development and testing development model and we're trying to change that to an agile way of building our systems. We are interfacing with the end users now, not banks, not airlines, not merchants but to the end users. we are changing the ways we do our things and number one is that we normally do discovery interviews and the discovery interviews is made with with the end users and trying to understand what challenges the face when they are making payments and what they want to have as a convenient way to make payments and then once you get all this feedback from the end users you do a usability test so we normally come up with a prototype like I think we have three or four iterations of Moula before we actually went live to the customers. what we do for usability tests is that actually we conduct AB testing where we give you a different experience for the same service, if you are going to buy go-tv then you have to go through a number of steps to get to go tv and let say that's option 1 than option 2 and option 3 are going to give you different steps but guy the same go tv. when we give this out to the customers we see how the customer interacts with the system, what challenges they have and we are able to say which one of the different option that is the best one to take and the other thing is that we usually go to an expert, either banks or we consult with other IT-experts. we come up with several experiences and they look at which of these experiences is the best and can we combine several experiences to make it an even better experience for the end user. That's what we have been trying to push to have the end user define what they want
65	Researcher:	Absolutely, that makes sense, very comprehensive. Maybe we can go into governance and control. What we have found exercising control over a platform meanwhile facilitating creative freedom of third-party development is of somewhat conflicting nature. what we're wondering is, with your experience from this company or previous ones, how do companies go about handling this balance. how do the discussion go in the conference-rooms when you're considering how to work with openness to both embrace the creative process and maintain control and security?
66	Rsp5:	There are two things that we need to maintain, number one is security and number 2 is the actual control, what the users do on the system. starting with the security, what we normally do in the system is that the third-party developers each have a set of credentials which means that even if you mess with the system, you're only messing with your own account and then the other thing is that we make sure that this is done over SSL which means that we have mutual identification that means that we can only speak to servers and services that we know. Mutual identification means that you're creating a pipe for each of the users that you are integrating to and this pipe is really secure. In terms of control, I mentioned a contract and we actually have two sets of contracts, data contracts we usually have with key merchants and these big companies. then we have like terms and conditions for smaller actors and individual developers. when you open the system to the third-party developers, we can have an endless amount of developers and it's hard to create a customized contract with each one of them. terms and conditions ensure that you don't mess with the system we have SSL connectivity, you have an account with us and for this account that you have with us it's an apprehended account, if you have the reversed where the customer have a postpaid model for integration to the system that means that their customers will be able to buy airtime from our system and then they emit the money to us. this is to prevent that a guy makes transactions for 10 million dollars and then they disappear on us. If your customers are going to consume 10 million credits then let's assign those credits and balance the float management and we have the transparency that you're able to check on how you consume that credit.
67	Researcher:	The last question that we have is, most of the time in integration environment the only communication you have with the customer is "the system is down etc", is there any other type of engaging communication, keeping customers engaged? are customers communicating with each other for an example?

68	Rsp5:	What we have with customers are account managers for our customers and the account managers have a set of customers that they interact with so the account managers are the ones who communicates to the customers any new improvements of the system, any changes, any additional services and are also those who take the customer feedback. they are getting feedback from the customers and they relay the same to the team leaders so we're able to come up with a way to improve the service. In addition to that we normally send out survey and this we normally do every quarter, this is a digital survey that we send out to our customers and they feel is good, the integration process are where do we have room for improvement. We normally say that the customers are always right so we really try our best to get feedback from the customers and the end users on what they expect the system to do for them, which we are able to translate into our host and our versatile system makes sure that the system is an important asset for them
69	Researcher:	We also have one question, given that you would start your own platform business, in a very hypothetical scenario, we have taken in these themes platform design, developer marketing and governance and security, is there any other consideration if you would imagine yourself being the one to pursue a business model built on third-party developers. Is there something that we have overlooked that is essential? I know in a previous interview for an example a guy mentioned talent acquisition
70	Rsp5:	I'm a bit confused, am I looking at it as the guy who s outsourcing the service or the guy is getting the outsourced service?
71	Researcher:	As a guy, you have the business that you want to be tapped into to outside innovation, you want other guys to build on your system for example. an example is the way that apple use APIs, people create developers to build apps. so you would be that guy, the platform host.
72	Rsp5:	Let's say I am outsourcing a service, I have the business and I want to outsource it, so what do I look for
73	Researcher:	Rather, how do I make myself marketable for someone else to use my service.
74	Rsp5:	The product and the service you provide, that would determine how much third parties want to work with you
75	Researcher:	So maybe we could say you have to provide a good value proposition
76	Rsp5:	Yes yes
77	Researcher:	That was very comprehensive and clear. Thank you very much
78	Rsp5:	You are welcome.
79	Researcher:	Goodnight
80	Rsp5:	Goodnight

Appendix 7: Interview Transcript [Rsp6]

Rsp6 = Respondent 6

Line	Person	Content
1	Rsp6::	Your study is of innovation in the context of external partners yes? So there are two things we do externally, we have this program which we call startpart, you can google it, what we do is that we invite startups and people that are basically already testing out the ideas and we invite them to the startpart program so startpart helps them test out new stuff or it could be like it's not a new invention but they're doing it in a different way that are looking to partner with us, then we find an opportunity to collaborate and scale up that business. That's one way, there's a whole different team that works with this at MasterCard that goes under the innovation team. Because we actually have a team that tries to drive innovation. the other we work with external parties, sometimes we outsource work with development, firms like Andela. But they work with something that is already existing, they don't do innovation from scratch
2	Researcher:	But, you have open APIs right?
3	Rsp6:	That's a way of also inviting, because we have the developer platform and we expose some of our APIs there so in that way we do partnerships. this company called flataway which is using some of our APIs and they also engage with us on our startpart program. So there's multiple ways, it's inviting outsiders through the startpart program and through the API platform though for us the API platform and startpart are both still work in progress. it's something that they are still building up, so there's the developer API, if you go to a platform is the experimental APIs and then there's also already established APIs, some of them are in banks and are being used, putting spend control on cards and things that are already out there.
4	Researcher:	Okay interesting, if we are to start on one of our first themes, and once again it's great that you exemplify but if you have prior experience that concern these questions we'll gladly soak that up as well, firstly it's developer motivations, if you have an open API, if you have an ecosystem, it's important of course to attract developers who wants to partner with you and there are different kinds of motivations and we have identified such things as access to educational tools, support, documentation, monetary incentives do you feel like there is any of these that stick out from your experience specifically that are more important to focus on?
5	Rsp6:	For us, our services are mostly financial and banking, so our technology is related to banking and finance so most of the people that are using this have innovations around the same area and I would argue that most of their motivations is around business and making money. maybe some of our new and experimental stuff will become more interesting for outside development as it becomes more advanced but it's not like people would learn just for the sake of it, that they don't have any legit business ideas that they want to actualize
6	Researcher:	And how do you support them in using your APIs to make money, you have this program for example that you just talked about. So I guess that would classify as an educational tool, do you offer other kinds of integration support, documentation?

7	Rsp6:	So we have actual team, an API-team that offers you all the support you need once your onboard so of course, since this is financial business we're very sensitive on partnerships and security, as much as the APIs are open, if you want to actualize this in a business idea, you need to go through someone who has been vetted liked partner so that means that it might take some time, but once you're onboard it is actually formal support, a whole team that supports you
8	Researcher:	So do you think that the market position of the company, based on its differentiation impacts the developers choice of platform
9	Rsp6:	I think that the developers choice of platforms relies on the market that they are targeting. because in paymarkets, different markets have different dominant players, so if you go to china, mastercard don't have much to say right? people will go for the wechat payment and all that. If you go to some countries in africa, you'll find that Visa is dominant. because also Visa has the APIs, so you realize that developers will therefore target based on their target market. We of course try to push our brand but we wouldn't really reason through from a brand dominance perspective unless we have something that is very useful for them that only we have
10	Researcher:	From that perspective, what do you do to evangelize to the developer community for instance, either in B2B contexts or events etc?
11	Rsp6:	I haven't attended one yet but we have constant events with the partners weather it is banks etc, because those who are driving the usage of these things are actually banks. because they are utilizing some of those APIs, so it will be mostly through banks. I mean, of course there is always an opportunity to expand that because I saw recently Visa sponsored a developer event in Kenya. people will keep expanding. maybe to a degree I'm still so new that I haven't seen it yet but we have not yet gone directly to the developers based on my knowledge, it's usually through the partners
12	Researcher:	When you talk about cards, you also have merchants that also integrate with you?
13	Rsp6:	Actually, the dominant model for a company like mastercard, is that even those merchants go through their partners and the majority of those partners are banks. The merchant would not typically go directly to us because what you see is that you end up having a lot of traffic, traffic that cannot handle in terms of support and all that. So if the partners handle that for you and then you deal only with the partner. then it becomes a bit more manageable.
14	Researcher:	Are you a hundred percent partner sold? or is it even a possibility to sell directly?
15	Rsp6:	In terms of our competitors, the only company who deals directly with merchants are AMEX. The rest of us tend to deal through partners
16	Researcher:	We're gonna move on to the next theme. when we have discussed with people before we talk alot about core business and losing your core business when you get too focused on building your ecosystem. So all in all, do you think a company should remain focused on your core business when opening APIs or when switching business models and trying to pursue this kind of business model or is it more important to try to create a healthy developer ecosystem and cater to everyone's needs?

17	Rsp6:	This has to do with the scaling. if you're gonna go directly to the market, which means that if you have a really good idea, then you can really go big, but the challenge becomes how you actually run support if there any issues. there are multiple ways of looking at it, but first to address the core business angle. because we are a technology company, the bigger you are the more you move out of your core business. As you grow bigger, you realize that everything is at risk of disruption so you also have to disrupt yourself
18	Researcher:	And diversify I presume?
19	Rsp6:	Yes exactly, that a few things so you don't focus only on your core business. I mean we won't do something crazy, like we won't go into farming, our core business is technology and we can do anything in technology and financial services although we are specialists in financial services. that drives the thinking. our model of going through partners is a just a way of managing scalability. although actors like AMEX have figured it out on how to go directly to consumers
20	Researcher:	So it's the scalability of the customer relationship management basicallys
21	Rsp6:	Yes especially because you are dealing globally. like how big are customer IKEA service, you need to handle so many people. I think it's an interesting area but it's a matter of choice at that point to be able to solve that problem
22	Researcher:	There's also this term that I'm sure you've heard before which is called critical mass, even from big incumbents who have started platform businesses. they've had extraordinary failures, google video for an example, when they weren't able to build a platform that had enough creators to attract viewers and not enough viewers to attract creators and what we are wondering is when starting of on this platform journey, even if you're an incumbent, how do you strategize penetrating the market and reaching a critical mass of user platform ecosystem attractive?
23	Rsp6:	I think the best way of selling a product is for the product to be like a compelling product, which is the hardest thing because sometimes you have a lot of marketing dollars that are spent, and that is one thing that is done definitely and you have all the engaging partners in events and supporting innovation which is a way of doing marketing. In the end of the day I think those who have really succeeded is those that have a really good product, and those are of course usually a minority of the products that you encounter. the other products are not good, but in terms of creating products that reach critical mass they have to be like really good. something that sells itself. I'm yet to experience it first hand in mastercard but how I'd like to think about it is that something that is really solving a paint point for someone, because sometimes you can spend a lot of marketing dollars, especially when you are dealing with developers. You don't really develop something that you are not so convinced about the case for it for you
24	Researcher:	Do you think you have to be able to sustain losses in the beginning? I guess that goes for all kind of product launches
25	Rsp6:	Yeah usually, if the business case is made for something within mastercard, as long as people are convinced, usually there's a readiness to sustain the losses
26	Researcher:	Let's assume for an example that you have a really differentiated product. in the end of the day people really needs to know about, needs to know how use it. is there any other strategies that go beyond having a really differentiated product

27	Rsp6:	For a company like us you triangulate a system, you that there are different departments, the person who came up with the innovation may be working closely with the people that develop but beyond that. you're pushing it to the market and you actually have to sell internally, someone to own it. you can't do everything you can't do marketing, the innovation, development and deployment all yourself so you actually have to sell it to someone internally and then from those people that you sell to internally are then going to be those people who then decide how they're going to market this for certain targets all that. so at least it's a sign of success if someone internally is convinced that you have something than you know that "yeah, you could be on to something good" and from there to come up with all the strategize to push this up. what's interesting for a company as big as mastercard, usually we're targeting a global audience with that you find that you don't have one expert on it there are probably different teams and tasks. so it's always an interesting process in how to roll things out
28	Researcher:	Exactly, the roll-out, the going live and launch phase is of course very crucial, especially when it comes to platforms. when you approach your APIs and releasing new ones, are you treating that as any other product launch? we've talked a little about marketing dollars.
29	Rsp6:	For us, API is part of a product. it wouldn't be on its own. It also depends on how the team that developed it is framing it. maybe you might have an API that is ready and that is extending another product and that is sold that way, but other than that. the API would be part of another product, so if you go to the API section you may find that something is interesting but you need something else for it to be actually useful, the other product
30	Researcher:	Speaking of boundary resources, in this case we refer to boundary resources as how your company has to relate to external actors and key ecosystem actors and the most common way of interfacing with the is SDKs and APIs and I'm just wondering if there's any aspects you take into consideration when designing such APIs? what is the priority, to you want to make very easy to use or? any specific consideration?
31	Rsp6:	Of course ease of use is important but what you find is that now also other things are coming into play like security, being a financial solution, security is number one. How secure is the solution. there are always people out there who are looking for vulnerabilities to attack, so whatever we design it has be secure and in terms of other product considerations, if it's going to eventually hit end customer or consumers, how do they interact with the solution, so it also has to be easy to use for the consumer. sometimes for the developer the API shouldn't be too painful to use but the priority first is the actual end consumer
32	Researcher:	In that same design when you target developers, is there any cases when they would be able to infringe on your core product? are there any architectural design considerations? you talked about vetting right of course
33	Rsp6:	Before we put anything out there, the security review that has been done is crazy enough. it's almost impossible that you find that something goes into compromise but of course you never say never. but it's something that is treated very seriously. What happens is that there are several layers, first the product that your aiming to develop is secure, the second is the context you're coming in, through which partner and that means that you are also limited further in what you are explicitly given access to. there's a product level security and then there's a partner level security in terms of what you've been given access to, so you're limited and also for you to come in as a partner there's certain checklists that you have to go through, in terms of development, if you just wanted to prove to a partner that you as a developer are able to implement something, then it would only be through the sandbox that is provided through the API-site. that's how you are able to do that.

34	Researcher:	Interesting, i mean is there any other resources you feel are beyond SDKs and APIs that platform hosts can use to engage and retain developers
35	Rsp6:	The beauty of the API level is that you limit the developer to a certain level of engagement
36	Researcher:	Exactly
37	Rsp6:	Just for that security thing. but you see what happens is if you rewind the scope and maybe go to google you see for them they have to make it easy because they cant go through that whole consideration, i donno partnerships etc they need developers to integrate directly because them they are playing a different game. If maybe they had to deal with money like we do, they might do it a bit differently. But i think because of the self service and also the standards that have come in i see the case as a big thing but also when you look at people like google they draw further engagement by having the google developer events, which people like MasterCard has not done, since that is mostly done by the likes of google , microsoft, they do big events to get people to try out they product and see how it works
38	Researcher:	Actually i come to realize that you have more of an arms length relationship as much as you have the apis
39	Rsp6:	Yes its still arms length just because of two considerations, how does that scale and then how do you protect the security of your partners coz definitely reputational risk is a very big thing with banks and financial institutions
40	Researcher:	When it comes to partners thou , even though they have to go through security and stuff they become a partner and they use your APIs , there was a time when mastercard did not have this, and when it comes to your own development processes , how this payment systems work for example, now you suddenly have to take into consideration like this partners and how they use their apis, do you think becoming a platform-based business has implied a change in how the internal development processes go , because i know we talked to an interviewee yesterday and he said it is extremely important that we don't make any changes to the core system that will affect our developers and our partners and how our developers work with our interface , do you feel there has been an impact there from becoming a platform-based company.

41	Rsp6:	<p>I think, for us we have not yet really identified as a platform company but its always about solving some of this problems with technology because i see if you are open up like google then you find you can really fie your growth because once you engage developers you can also really do alot of innovation for you, otherwise it has to be kind of hard for new things to come through. definitely you can see there is some limitation there where we can do better if we are able to open it up , so i think its maybe a question of engaging sometimes with the security team and convincing them that you are not going to bring down the company with your choice because if you have dealt with people that work with security , they are like accountants they don't change easily. you have to really work hard to convince them that if you are changing the game it actually makes sense and the company is still protected , but i am definitely for opening up and changing strategy to become more of an open platform. I think that would be great but however will really crack that would need to tackle things like support coz you know with support you can partner with someone like a company to provide support and say that this is going to be a big deal just outsource this and provide a lot of online tools, but the whole question of security from the human aspect , like for example mpesa in kenya you could argue is secure but the human element of people actually using that and using social engineering to defraud people and that kind of thing, which is now the protection of partnerships bringin because they are able to do KYC and all that coz and online platform that stands on its own you are only able to do so much KYC. and that works for google because the API they are giving you has to do with maps and a wide array of things where they don't really need to know you that intimately but when it comes to you are dealing will peoples money then the bar changes coz it means you have to protect the reputational risk that would come from people saying , beware of mastercard products , there are few fraudsters that deal with that platform, you might get de-coded . you don't want that reputational risk</p>
42	Researcher:	<p>but now then when you are making changes to the apis, i presume the partners are well communicated to far in advance so they have the to make the changes they need to do on their part for example there is a new functionality or new interface coming out to make sure they dont need to rebuild their entire website</p>
43	Rsp6:	<p>Yeah that one is there and we practice say like the versioning because let's say you have an api that is out there and its being used, then there is a whole management aspect of versioning of transitioning to new versions without breaking because some of this banks do not have developers on standby or if they have developers they are busy on other things so you don't want to disrupt them, so the work of the api is to make sure those transitions are done and you don't have any downtime even during maintenance or upgrading because definitely for banks there are some people that work with old versions of tech because for them is also very expensive to change or roll out some of this products</p>
44	Researcher:	<p>So is the strategy then to make as minor changes as possible you think</p>
45	Rsp6:	<p>Is to whatever changes are made to not impact previous versions, so its like you post the new version and that stands on its own and the old version is still working with full access to its functionality so unless there is a security issue, there wouldn't be a case where you want to force people to upgrade because the banks just won't do it in our case, and its the same thing when you look at maybe like facebook and some of those that give apis , they also take a very long time to actually sunset their api. like i know a long time ago when i still used to do dev work i implemented some old apis and many years later is when they implemented , so it has to be necessitated by things like security and its very important that the old version stops working. coz you find some of the banks are still on windows xp and the like so they take very long to make changes so you have to be sensitive to that which is the challenge of once you decide to open up and have many customers using your API.</p>

46	Researcher:	Well, is there any specific rules of engagement that come to mind. I know you have your terms and conditions when you engage with external developers in order to maintain trust, there is the pricing model, is there anything else that stands out in how you engage with external development
47	Rsp6:	Security review, and you know pricing that is a commercial concern since its a willing buyer, willing seller . The other side is about what you are integrating, what is it doing , how is it affecting us in terms of load
48	Researcher:	Okay so just a finishing question from me. What I find particularly interesting about mastercard is that we are looking at how organizations struggle to balance control of how open they should be and I think mastercard is a clear example of a very controlled infrastructure, as much as you still have a bunch of APIs, what would your opinion on how to really exercise control over a platform and at the same time facilitate efficient control over third-party development.. I mean this must be very interesting for you of course because there must be constant reviews where people sit in cool and big conference rooms and discuss if we should open or should we keep it closed and how do we strategize these things.
49	Rsp6:	There are two levels, there's access to the ideas for development which is mostly open, where you have access to a sandbox but you the challenge is how do I actualize this in the real life environment? that's where it becomes difficult, at least from us and our perspective. so we try to gather as much ideas as possible but you know this would be driven by the product teams, like how do we want to open this, for us and if we went to the API site for mastercard, about 80 percent of the APIs I can't tell you what they do because they are driven by a specific product team so they would be the ones at that small level that are determining on how to approach this and it's ore of an issue of scaling, how successful and relevant is this technology for being opened up, how much do you expect to gain from opening it up and then here's the sandbox that you have given people for development, but in terms of onboarding developers through the partner, that becomes a whole different question, the problem is that you become a platform-based community like that you have to engage with the deployment teams and the people on the ground that are actually engaging with the banks because I know this as well maybe they disconnect easier, because the person deploying it most likely is not the person that owned the product from the start so you really have to engage closely to make sure that you don't lose touch completely
50	Researcher:	There must be constant risk assessment then?
51	Rsp6:	Yeah, but at the product level so the product with technology, but usually at that point even if you're talking lime production staff, like for me I don't touch production. that's a whole different group with its own sensitivities, they are very paranoid
52	Researcher:	Does this means that this sandbox approach only applies to mastercard?

53	Rsp6:	I haven't looked at Visa but I believe that they are doing the same. you know the only way you can really test is through a sandbox. the only way to test and figure out if things actually work. it's a consolation for the developers, here's how you can figure out how it works. the beauty with firms with google is that you get your interact keys and then you're up and running which is a big different, the issue with like mastercard, because we are dealing with money mostly. so now, where is that money coming from, if it's coming for let's say your bank, have your bank allowed you to interact with us in that way, because if you could just open an interface so easy, you'd find a situation whereby maybe a conman creates an app an he puts a lot of stuff to validate his con on his application and then through the internet, through whatsapp or whatever distributes them, that's why there's a very big difference in being on the sandbox and being in production. I think this is really unique to acknowledge if you're dealing with financial services solution, it may not be the same case with some of the others
54	Researcher:	We have talked to very different interviewees and what's interesting is that there are a lot of considerations that are the same but of course since security is so important for you, it makes everything a bit more special. I was thinking about a more hypothetical question now that may now relate that much to your position at mastercard as uch, let's say that you were in a situation where you were gonna launch your own platform. You can put in whatever premises you'd like and then the question is that we have these themes right, developer marketing, platform design and governance and control, do you feel that there's anything else that pops up on the top of your head that you would also take into consideration, anything that we have overlooked you would consider crucial in a platform-based business?
55	Rsp6:	Definitely for me if I was looking at it hypothetically, opening it up would really help. If I open it up I may have then it means that I have a bigger chance of success with scale. You see for me, I'm different, I don't have the number of partners that mastercard has. I'm coming from a startup position right. It has to be about engaging developers and also my risk capital is different because I don't have the brand to risk. maybe ones I get a few problems and of course I have to wait based on the risk factors. It's not that I'm not sensitive on security but I would put as much weight on it. what I might do in this case is my own assessment of the developers that are coming on and just evaluating what they want to do and how they are doing it and then monitoring that. I might therefore include a lot more human intervention because you see, in small scale and as a startup human intervention is possible still has the capacity to look at it, but when you are mastercard with billions things happening on your system, you're not able to. personally, opening it up as much as possible and allowing developers to selfserve and then I just monitor in the back.
56	Researcher:	When you look at this on a scale on how open someone is from linux to, now I feel like mastercard is almost on the other end.
57	Rsp6:	When you look at the developer portal, it's really something that is changing. something that people have changed but in a measure it's all about bit by bit and seeing how much we can open because definitely the world is becoming much more open but even as you are doing it you have to be aware on what else is happening out there because of our use cases. But if we had a product for an example that was not concerning money for an example, hypothetically, I would imagine everything would be much more open and much more self serve.
58	Researcher:	Despite the business model and the product. do you feel like the way you design, govern and market to developer. do you feel that it is anything different on how you would build and govern a very open and a very closed system or is it just the onboarding technicalities that are different

59	Rsp6:	I actually can't tell you specifically if we have developer marketing or not. I don't know that per se. the challenge is in a big company like mastercard with twelve thousand people, someone could be doing something that you have no idea about and you may end up giving statement that and then find that the exact opposite is happening. This is just from my own experience, only. But we're not so busy when it comes to that type of marketing when you compare to the likes of microsoft and all those. they're definitely doing much more in terms of engaging developers. So, i think that the more open the platform, the more I have to go and reach out weather it's in campuses, if you look at a company like africa is talking, who do the sms products, they actually go to the level of engaging students in campuses, giving them those tools and those ideas meanwhile making sure that they are the the top of their mind of those developers from early on. meaning that the developers choice would be africa is talking for sms integration. I think that's where the key thing is in terms of how aggressively do you engage but you see also for us, if most of our APIs are financial and we are mostly integrating developers, that means unless equity bank or someone like that decided let's drive this with mastercard and then see what developers come and do then you see that it's a whole different ball game. Because you can't go to a campus and say, we have all these APIs and then they ask you, so how do we develop on it? and you have to give them this whole longlist of things they have to do
60	Researcher:	Okay, thank you so much for you time Edwin!
61	Rsp6:	Thank you for also giving me some food for thought in this area
62	Researcher:	Bye!
63	Rsp6:	Bye!

References

- Accenture. (2018). *Engaging the Developer Community: What Developer Ecosystems Need to Know*. Retrieved from https://www.accenture.com/t20180202T092215Z_w_us-en/acnmedia/PDF-70/Accenture-Digital-Ecosystems-POV.pdf#zoom=50
- Alstyne, M. W. V., Parker, G. G., & Choudary, S. P. (2016). Pipelines, Platforms, and the New Rules of Strategy. *Harvard Business Review*.
- Altman, E. J., & Tripsas, M. (2014). *Product to Platform Transitions: Organizational Identity Implications*. Oxford Handbook of Creativity, Innovation, and Entrepreneurship: Multilevel Linkages. Oxford, UK: Oxford University Press.
- Basole, R. C. (2009). Visualization of interfirm relations in a converging mobile ecosystem. *Journal of Information Technology*, 24(2), pp. 144-159.
- Bender, B., & Gronau, N. (2017). *Coring on Digital Platforms – Fundamentals and Examples from the Mobile Device Sector*. Paper presented at the Thirty Eighth International Conference on Information Systems Proceedings, South Korea.
- Bengtsson, M., & Kock, S. (2000). 'Coopetition' in Business Networks : to Cooperate and Compete Simultaneously. *Industrial Marketing Management*, 29(5), pp. 411-426.
- Benlian, A., Hilkert, D., & Hess, T. (2015). How open is this platform? The meaning and measurement of platform openness from the complementors' perspective. *Journal of Information Technology*, 30(3), pp. 209-228.
- Benner, M. J. (2007). The incumbent discount: Stock market categories and response to radical technological change. *Academy of management review*, 32(3), pp. 703-720.
- Benzell, S. G., Lagarda, G., & Alstyne, M. V. (2017). The Impact of APIs on Firm Performance. Retrieved from SSRN website: <https://ssrn.com/abstract=2843326>
- Bergman, M., Lyytinen, K., & Mark, G. (2007). Boundary Object in Design: An Ecological View of Design. *Journal of the Association of Information Systems*, 8(11), pp. 546-568.
- Bhargava, H. K., & Choudhary, V. (2004). Economics of an information intermediary with aggregation benefits. *Information systems research*, 15(1), pp. 22-36.
- Bhattacharjee, A. (2012). *Social science research: Principles, methods, and practices*. Open Access Textbooks Collection. Book 3.
- Bianco, V. D., Myllarniemi, V., Komssi, M., & Raatikainen, M. (2014). *The Role of Platform Boundary Resources in Software Ecosystems: A Case Study*. Paper presented at the Software Architecture (WICSA), 2014 IEEE/IFIP Conference, Sydney, NSW, Australia.
- Boland, R. J., Lyytinen, K., & Yoo, Y. (2007). Wakes of innovation in project networks: The case of digital 3-D representations in architecture, engineering, and construction. *Organization Science*, 18(4), pp. 631-647.
- Boudreau, K. (2010). Open platform strategies and innovation: Granting access vs. devolving control. *Management science*, 56(10), 1849-1872.
- Boudreau, K., & Hagiu, A. (2009). Platform rules: Multi-sided platforms as regulators. *Platforms, markets and innovation*, 1, pp. 163-191.
- Boudreau, K., & Lakhani, K. (2009). How to manage outside innovation. *MIT Sloan Management Review*, 50(4), pp. 69-76.
- Boudreau, K. J., & Lakhani, K. R. (2009). How to Manage Outside Innovation. *MIT Sloan Management Review*, 50(4), pp. 69-76.

- Bresnahan, T. F., & Levin, J. D. (2012). *Vertical integration and market structure*. Retrieved from <http://www.nber.org/papers/w17889>
- Bryman, A., & Bell, E. (2011). Ethics in business research. *Business Research Methods*.
- Bryman, A., & Bell, E. (2015). *Business research methods*: Oxford University Press, USA.
- Budac, C., & Baltador, L. (2013). The value of brand equity. *Procedia Economics and Finance*, 6, pp. 444-448.
- Bui, Q. (2015). *A review of innovation diffusion theories and mechanisms*. Paper presented at the Diffusion Interest Group In Information Technology Proceedings 2015.
- Cabrera, M. (2014). Use Co-opetition to build new lines of Revenue. *Harvard Business Review*, 10.
- Ceccagnoli, M., Forman, C., Huang, P., & Wu, D. (2012). Cocreation of Value in a Platform Ecosystem! The Case of Enterprise Software. *MIS quarterly*, pp. 263-290.
- Chesbrough, H. (2007). Business model innovation: it's not just about technology anymore. *Strategy & leadership*, 35(6), pp. 12-17.
- Chesbrough, H., Vanhaverbeke, W., & West, J. (2006). *Open Innovation: Researching a New Paradigm*: Oxford University Press, Oxford.
- Chesbrough, H. W., & Appleyard, M. M. (2007). Open Innovation and Strategy. *California Management Review*, 50(1), pp. 57-76.
- Clever, N., Holler, J., Püster, J., & Shitkova, M. (2013). *Growing Trees-A Versioning Approach For Business Process Models Based On Graph Theory*. Paper presented at the ECIS.
- Dellermann, D., Jud, C., & Reck, F. (2017). Understanding platform loyalty in the cloud: a configurational view on ISV's costs and benefits.
- Diener, E., & Crandall, R. (1978). *Ethics in social and behavioral research*: U Chicago Press.
- Diller, S., Shedroff, N., & Rhea, D. (2005). *Making meaning: How successful businesses deliver meaningful customer experiences*: New Riders Publishing.
- DJI. (2018). DJI SDK - Develop solutions for different industries upon your needs. Retrieved from <https://developer.dji.com/>
- Eaton, B., Elaluf-Calderwood, S., Sørensen, C., & Yoo, Y. (2015). Distributed Tuning of Boundary Resources: The Case of Apple's iOS Service System. *MIS Quarterly*, 39(1), pp. 217-243.
- Eisenmann, T. R., Parker, G., & Van Alstyne, M. W. (2010). Platform envelopment.
- Eisenmann, T. R., Parker, G. G., & Alstyne, M. W. V. (2006). Strategies for Two-Sided Markets. *Harvard Business Review*.
- Evans, D. S. (2009). How catalysts ignite: the economics of platform-based start-ups. *Platforms, markets and innovation*, 99-128.
- Evans, D. S. (2016). *PLATFORM ECONOMICS: Essays on Multi-Sided Businesses*: Competition Policy International (CPI).
- Evans, D. S., Hagiu, A., & Schmalensee, R. (2006). *Invisible Engines: How Software Platforms Drive Innovation and Transform Industries*: The MIT Press, Cambridge, Massachusetts.
- Evans, D. S., & Schmalensee, R. (2010). Failure to launch: Critical mass in platform businesses. *Review of Network Economics*, 9(4).
- Evans, P. C., & Gawer, A. (2016). *The Rise of the Platform Enterprise: A Global Survey*. Retrieved from The Center for Global Enterprise: <http://thecge.net/>
- Feller, J., & Fitzgerald, B. (2002). *Understanding open source software development*: Addison-Wesley London.
- Flick, U. (2014). *The SAGE Handbook of Qualitative Data analysis*. SAGE Publications Ltd.

- Fortt, J. (2006). Cost of failure for Google Video: \$1.65 billion. Retrieved from <http://fortune.com/2006/10/09/cost-of-failure-for-google-video-1-65-billion/>
- Free, D. (2018). New Accenture Research Reveals Platform Providers Face Gap in Meeting Software Developers' Expectations. Retrieved from <https://newsroom.accenture.com/news/new-accenture-research-reveals-platform-providers-face-gap-in-meeting-software-developers-expectations.htm>
- Gandel, S. (2016). These Are the 10 Most Valuable Companies in the Fortune 500. Retrieved from <http://fortune.com/2016/02/04/most-valuable-companies-fortune-500-apple/>
- Gawer, A., & Cusumano, M. A. (2002). *Platform leadership: How Intel, Microsoft, and Cisco drive industry innovation* (Vol. 5): Harvard Business School Press Boston.
- Gawer, A., & Cusumano, M. A. (2014). Industry Platforms and Ecosystem Innovation. *Journal of Product Innovation Management*, 31(3), pp. 417–433. doi:10.1111/jpim.12105
- Ghazawneh, A., & Henfridsson, O. (2010). *Governing Third-Party Development Through Platform Boundary Resources*. Paper presented at the Thirty First International Conference on Information Systems, (ICIS) 2010 Proceedings, St. Louis. http://aisel.aisnet.org/icis2010_submissions/48
- Ghazawneh, A., & Henfridsson, O. (2013). Balancing platform control and external contribution in third-party development: the boundary resources model. *Information Systems Journal*, 23(2).
- Ghazawneh, A., & Henfridsson, O. (2015). A paradigmatic analysis of digital application marketplaces. *Journal of Information Technology*, 30(3), pp. 198-208.
- Glasser, B., & Strauss, A. (1967). The development of grounded theory. *Chicago, IL: Alden*.
- Glynn, M. A., & Abzug, R. (2002). Institutionalizing identity: Symbolic isomorphism and organizational names. *Academy of Management Journal*, 45(1), pp. 267-280.
- Goldbach, T., & Benlian, A. (2015a). *How Informal Control Modes affect Developers' Trust in a Platform Vendor and Platform Stickiness a Platform Vendor and Platform Stickiness*. Paper presented at the Twelfth International Conference on Wirtschaftsinformatik, Osnabrück.
- Goldbach, T., & Benlian, A. (2015b). *Understanding Informal Control Modes on Software Platforms – The Mediating Role of Third-Party Developers' Intrinsic Motivation*. Paper presented at the Thirty Sixth International Conference on Information Systems, Fort Worth.
- Goldbach, T., & Kemper, V. (2014). *Should I Stay or Should I Go? The Effects of Control Mechanisms on App Developers' Intention to Stick with a Platform*. Paper presented at the Twenty Second European Conference on Information Systems, Tel Aviv.
- Grunert, K. G., & Grunert, S. C. (1995). Measuring subjective meaning structures by the laddering method: Theoretical considerations and methodological problems. *International journal of research in marketing*, 12(3), 209-225.
- Hagedoorn, J. (1993). Understanding the rationale of strategic technology partnering: Nterorganizational modes of cooperation and sectoral differences. *Strategic Management Journal*, 14(5), pp. 371-385.
- Hagiu, A., & Rothman, S. (2016). Network Effects Aren't Enough: The hidden traps in building an online marketplace. *Harvard Business Review*, pp. 65-71.
- Hart, S., & Tzokas, N. (2000). New product launch “mix” in growth and mature product markets. *Benchmarking: An International Journal*, 7(5), pp. 389-405.
- Hasselbring, W., & Reussner, R. (2006). Toward trustworthy software systems. *Computer*, 39(4), pp. 91-92.

- Helm, I. M. R., Holland, D., & Gangopadhyay. (1990). *Contracts: Specifying Behavioral Compositions in Object-Oriented Systems*. Paper presented at the Proceedings OPPSLA/ECOOP, Ottawa, Canada.
- Hemetsberger, A., & Reinhardt, C. (2004). *Sharing and creating knowledge in open-source communities: The case of kde*. Paper presented at the Paper for Fifth European Conference on Organizational Knowledge, Learning, and Capabilities, Innsbruck.
- Hemetsberger, A., & Reinhardt, C. (2006). Learning and knowledge-building in open-source communities: A social-experiential approach. *Management learning*, 37(2), pp. 187-214.
- Hobcraft, P. (2017). Platform models are the new order of business need. Retrieved from <https://blog.hypeinnovation.com/platform-models-are-the-new-order-of-business-need>
- Holcomb, T. R., & Hitt, M. A. (2007). Toward a model of strategic outsourcing. *Journal of operations management*, 25(2), pp. 464-481.
- Hsieh, J., & Hsieh, Y. (2013). Appealing to Internet-based freelance developers in smartphone application marketplaces. *International Journal of Information Management*, 33(2), pp. 308-317.
- Hyrnsalmi, S., Suominen, A., & Mäntymäki, M. (2016). The influence of developer multi-homing on competition between software ecosystems. *Journal of Systems and Software*, 111, 119-127. doi:10.1016/j.jss.2015.08.053
- Iansiti, M., & Levien, R. (2004). *The Keystone Advantage: What the New Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability*: Harvard Business School Press: Boston, MA.
- Jansen, S., Brinkkemper, S., Finkelstein, A., & Bosch, J. (2009). *Introduction to the Proceedings of the First Workshop on Software Ecosystems*. Paper presented at the IWSECO@ ICSR.
- Kankanhalli, A., Ye, J. H., & Teo, H. H. (2015). Comparing Potential and Actual Innovators: An Empirical Study of Mobile Data Services Innovation. *MIS quarterly*, 39(3), pp.667-682.
- Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS quarterly*, 23(1), pp. 67-94.
- Kogut, B., & Zander, U. (1996). What firms do? Coordination, identity, and learning. *Organization science*, 7(5), pp. 502-518.
- Kouzmin, A., & Korac-Kakabadse, N. (2000). Mapping institutional impacts of lean communication in lean agencies: information technology illiteracy and leadership failure. *Administration & Society*, 32(1), pp. 29-69.
- Kude, T., Dibbern, J., & Heinzl, A. (2012). Why do complementors participate an analysis of partnership networks in the enterprise software industry. *IEEE Transactions on Engineering Management*, 59(2), pp. 250-265.
- Kvale, S. (1996). The 1,000-page question. *Qualitative inquiry*, 2(3), pp. 275-284.
- Kvale, S. (2006). Dominance through interviews and dialogues. *Qualitative inquiry*, 12(3), pp. 480-500.
- Kvale, S., & Brinkmann, S. (2009). *Interviews: Learning the craft of qualitative research*. California, US: SAGE, pp. 230-243.
- Lacity, M. C., & Johnson, M. A. (1994). Understanding qualitative data: A framework of text analysis methods. *Journal of Management Information Systems*, 11(2), pp. 137-155.
- Laffan, L. (2011). *Open Governance Index: Measuring the true openness of open source projects from Android to Webkit*. Retrieved from www.visionmobile.com/blog:
- Lakhani, K. R., & Von Hippel, E. (2003). How open source software works: "free" user-to-user assistance. *Research policy*, 32(6), pp. 923-943.

- Lee, A. S., & Baskerville, R. L. (2003). Generalizing generalizability in information systems research. *Information systems research*, 14(3), 221-243.
- Leijon, E., Svenheden, J., & Svahn, F. (2017). *Platform Thinking in Incumbent Firms: From Concept to Capability*. Paper presented at the Proceedings of the 50th Hawaii International Conference on System Sciences.
- Lewis, J., & Ritchie, J. (2003). Generalising from qualitative research. *Qualitative research practice: A guide for social science students and researchers*, 2, 347-362.
- Lin, M., Li, S., & Whinston, A. B. (2011). Innovation and Price Competition in a Two-Sided Market. *Journal of Management Information Systems*, 28(2), pp. 171-202.
- Mankevich, V. (2014). Managing Innovation Networks: Exploring Coopetition Dynamics in Innovation Ecosystems.
- Manner, J., Nienaber, D., Schermann, M., & Krmar, H. (2013). *Six Principles for Governing Mobile Platforms*. Paper presented at the Wirtschaftsinformatik Proceedings 2013.
- Mastercard. (2018). Updates & ideas from Mastercard Developers. Retrieved from <https://developers.mastercard.com/apis>
- McIntyre, D. P., & Srinivasan, A. (2017). Networks, platforms, and strategy: Emerging views and next steps. *Strategic Management Journal*, 38(1), pp. 141-160.
- Midha, V., & Bhattacharjee, A. (2012). Governance practices and software maintenance: A study of open source projects. *Decision Support Systems*, 54(1), 23-32.
- Miles, M. B., & Huberman, A. M. (1984). Drawing valid meaning from qualitative data: Toward a shared craft. *Educational researcher*, 13(5), pp. 20-30.
- Miles, M. B., Huberman, A. M., & Saldana, J. (2014). *Qualitative data analysis: A method sourcebook*. CA, US: Sage Publications.
- Mohagheghzadeh, A., & Svahn, F. (2016). *Shifting Design Capability to Third-Party Developers: An Affordance Perspective on Platform Boundary Resources*. Paper presented at the Twenty-second Americas Conference on Information Systems, San Diego.
- Moore, J. F. (1996). *The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems*.: HarperBusiness: New York, NY.
- Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and Organization*, 17, pp. 2-26.
- Nocke, V., Peitz, M., & Stahl, K. (2007). Platform ownership. *Journal of the European Economic Association*, 5(6), pp. 1130-1160.
- Nylén, D., & Holmström, J. (2015). Digital innovation strategy: A framework for diagnosing and improving digital product and service innovation. *Business Horizons*, 58(1), 57-67. doi:10.1016/j.bushor.2014.09.001
- O'Grady, S. (2013). *The New Kingmakers: How Developers Conquered the World*: O'Reilly Media.
- O'Mahony, S., & Ferraro, F. (2007). The emergence of governance in an open source community. *Academy of Management Journal*, 50(5), pp. 1079-1106.
- Oh, J., Koh, B., & Raghunathan, S. (2015). Value appropriation between the platform provider and app developers in mobile platform mediated networks. *Journal of Information Technology*, 30(3), pp. 245-259. doi:<https://doi.org/10.1057/jit.2015.21>
- Parker, G., Alstyne, M. V., & Jiang, X. (2017). Platform Ecosystems: How Developers Invert the Firm. *MIS quarterly*, 41(1), pp. 255-266.
- Perrons, R. K. (2009). The open kimono: How Intel balances trust and power to maintain platform leadership. *Research Policy*, 38(8), pp. 1300-1312.
- Pisano, G. P., & Verganti, R. (2008). Which kind of collaboration is right for you. *Harvard Business Review*, 86(12), pp. 78-86.

- Popp, K., & Meyer, R. (2010). *Profit from Software Ecosystems: Professional edition, Business Models, Ecosystems and Partnerships in the Software Industry*. Books on Demand GmbH.
- Recker, J. (2013). *Scientific research in information systems: a beginner's guide*. Springer Heidelberg New York Dordrecht London.
- Reuve, M. d., Sørensen, C., & Basole, R. C. (2017). The digital platform: a research agenda. *Journal of Information Technology*, pp. 1-12. doi:<https://doi.org/10.1057/s41265-016-0033-3>
- Reynolds, T. J., & Gutman, J. (1988). Laddering theory, method, analysis, and interpretation. *Journal of advertising research*, 28(1), 11-31.
- Rickmann, T., Wenzel, S., & Fischbach, K. (2014). *Software Ecosystem Orchestration: The Perspective of Complementors*. Paper presented at the Twentieth Americas Conference on Information Systems , 2014, Savannah.
- Rogers, E. M. (2010). *Diffusion of Innovations* (4th ed.): The Free Press. Simon & Schuster, New York.
- Sagers, G. (2004). The influence of network governance factors on success in open source software development projects. *ICIS 2004 Proceedings*, 34.
- Sarkhel, A. (2017). Developers are the future decision makers: IBM's Tejada. Retrieved from <https://tech.economictimes.indiatimes.com/news/people/developers-are-the-future-decision-makers-ibms-tejada/59896321>
- Schilling, M. (2009). *Protecting or using a technology platform: Trade offs in appropriability, network externalities and architectural control*. In A. Gawer (Ed.), *Platforms, markets and innovation*. Cheltenham, UK: Edward Elgar Publishing.
- Schreieck, M., Wiesche, M., & Krcmar, H. (2016). *Design and Governance of Platform Ecosystems: Key Concepts and Issues for Future Research*. Paper presented at the Twenty-Fourth European Conference on Information Systems (ECIS), İstanbul, Turkey.
- Schuermans, S., Vakulenko, M., & Constantinou, A. (2015). *IOT Megatrends 2016: Six key trends in the IoT developer economy*. Retrieved from Vision Mobile: <https://www.slashdata.co/reports/iot-megatrends-2016>
- Schuermans, S., Wilcox, M., Hecht, L., & Voskoglou, C. (2017). *Developer Economics: State of the Developer Nation Q3 2017*. Retrieved from SlashData: <https://www.slashdata.co/reports/state-of-the-developer-nation-q3-2017>
- Schultze, U., & Avital, M. (2011). Designing interviews to generate rich data for information systems research. *Information and Organization*, 21(1), pp. 1-16.
- Seale, C. (1999). Quality in qualitative research. *Qualitative inquiry*, 5(4), 465-478.
- Seawright, J., & Gerring, J. (2008). Case selection techniques in case study research: A menu of qualitative and quantitative options. *Political Research Quarterly*, 61(2), 294-308.
- Selander, L., Henfridsson, O., & Svahn, F. (2013). Capability search and redeem across digital ecosystems. *Journal of Information Technology*, 28(3), pp. 183-197.
- Shah, S. K. (2006). Motivation, governance, and the viability of hybrid forms in open source software development. *Management science*, 52(7), pp. 1000-1014.
- Shuradze, G., Wagner, D., & Wagner, H. T. (2015). *Technological Platforms and Innovation: Review, Integration, and Extension*. Paper presented at the Twenty-first Americas Conference on Information Systems (AMCIS), Puerto Rico.
- Smedlund, A., & Faghankhani, H. (2015). *Platform Orchestration for Efficiency, Development and Innovation*. Paper presented at the Forty Eighth Hawaii International Conference on System Sciences.

- Sordi, J. O. D., Nelson, E., Meireles, M., & Silveira, M. A. D. (2016). Development of digital products and services: Proposal of a framework to analyze versioning actions. *European Management Journal*, 34(5), pp. 564-578.
- Star, S. L., & Griesemer, J. R. (1989). Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39. *Social Studies of Science*, 19(3), pp. 387-420.
- Staudenmayer, N., Tripsas, M., & Tucci, C. L. (2005). Interfirm modularity and its implications for product development. *Journal of Product Innovation Management*, 22(4), pp. 303-321.
- Staykova, K., & Damsgaard, J. (2015). *A Typology of Multi-sided Platforms: The Core and the Periphery*. Paper presented at the Twenty Third European Conference on Information Systems, Münster.
- Tiwana, A. (2013). *Platform ecosystems: aligning architecture, governance, and strategy*: Newnes.
- Tiwana, A. (2014). *Platform Ecosystems: Aligning Architecture, Governance, and Strategy*: Morgan Kaufmann.
- Tiwana, A. (2015). Evolutionary Competition in Platform Ecosystems. *Information systems research*, 18(1), pp. 7047-7047.
- Tiwana, A., Konsynski, B., & Bush, A. A. (2010). Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics. *Information systems research*, 21(4), pp. 675-687.
- Top, S., Dilek, S., & Çolakoglu, N. (2011). Perceptions Of Network Effects: Positive Or Negative Externalities? *Procedia Social and Behavioral Sciences*, 24, pp. 1574-1584.
- Triche, J., Cao, Q., & Thompson, M. (2013). Exploring the impact of online reviews with brand equity for online software purchasing behavior.
- Tura, N., Kutvonen, A., & Ritala, P. (2017). Platform Design Framework: Conceptualisation and Application. *Technology Analysis & Strategic Management*. doi:10.1080/09537325.2017.1390220
- Twilio. (2018). Twilio Pricing: Only pay for what you use. Don't get stalled by contracts, capacity planning, and price modeling. Retrieved from <https://www.twilio.com/pricing>
- Uber. (2018). Introduction to Affiliate Program. Retrieved from <https://developer.uber.com/docs/riders/affiliate-program/introduction>
- Um, S., & Yoo, Y. (2016). *The Co-Evolution of Digital Ecosystems*. Paper presented at the Thirty Seventh International Conference on Information Systems, Dublin.
- Vakulenko, M. (2016). 5 ways developers can extend your business model. Retrieved from <https://www.slashdata.co/blog/2016/01/5-ways-developers-can-extend-your-business-model>
- Walgreens. (2018). Print it, share it: Take memories beyond your camera roll. Retrieved from <https://photo.walgreens.com/store/prints?tab=Photo%20Nav%20>
- Walsham, G. (2006). Doing interpretive research. *European journal of information systems*, 15(3), pp. 320-330.
- Van de Ven, A. H. (2005). Running in packs to develop knowledge-intensive technologies. *MIS quarterly*, pp. 365-377.
- Van de Ven, A. H., Polley, D., Garud, R., & Venkatraman, S. (2008). *The Innovation Journey*: New York: Oxford University Press.
- Wan, X., Cenamor, J., Parker, G., & Alstyne, M. V. (2017). Unraveling Platform Strategies: A Review from an Organizational Ambidexterity Perspective. *Sustainability*, 7(734). doi:10.3390/su9050734

- Vannieuwenborg, F., Mainil, L., Verbrugge, S., Pickavet, M., & Colle, D. (2012). *Business models for the mobile application market from a developer's viewpoint*. Paper presented at the 16th International Conference on Intelligence in Next Generation Networks (ICIN).
- Weill, P., & Woerner, S. L. (2013). Optimizing Your Digital Business Model. *MIT Sloan Management Review*, 54(3), pp. 71-78.
- Weinreich, R., Ziebermayr, T., & Draheim, D. (2007). *A versioning model for enterprise services*. Paper presented at the Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on 2007.
- Winter, S. G. (2003). Understanding dynamic capabilities. *Strategic Management Journal*, 24(10), pp. 991-995.
- Von Hippel, E., & Katz, R. (2002). Shifting innovation to users via toolkits. *Management science*, 48, pp. 821–834.
- Yin, R. (2003). Case study research: Design and methods. *Sage Publications, Inc*, 5, 11.
- Yoo, Y., Lyytinen, K., & Boland, R. J. (2008). *Distributed Innovation in Classes of Networks*. Paper presented at the Proceedings of the Fourty-First Annual Hawaii International Conference on System Sciences.
- Yoo, Y., Lyytinen, K., & Boland, R. J. (2009). *Innovation in the Digital Era: Digitization and Four Classes of Innovation Networks*. Working Paper, Temple University.
- Youngjin, Y., Ola, H., & Kalle, L. (2010). The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research. *Information systems research*, 21(4), pp. 724–735.
- Zhu, F., & Furr, N. (2016). Products to Platforms: Making the Leap. *Harvard Business Review*, pp. 75-78.
- Zhu, F., & Iansiti, M. (2012). Entry into platform-based markets. *Strategic Management Journal*, 33(1), pp. 88-106.