Relationship governance in mobile ecosystems

The case of mobile application developers

Master Thesis – Management of Technology

Faculty of Technology, Policy, and Management – Delft University of Technology

Roel Hakvoort

1237632

August, 2013
Graduation Committee

**Formal chair**
Prof. dr. C.P. van Beers
Full professor
Delft University of Technology
Section Technology, Strategy, and Entrepreneurship

**First supervisor**
Dr. E. den Hartigh
Assistant professor
Delft University of Technology
Section Technology, Strategy, and Entrepreneurship

**Second supervisor**
Dr. M. de Reuver
Assistant professor
Delft University of Technology
Section Information and Communication Technology
Preface
The report before you is the end result of my education in Delft. A master’s thesis for Management of Technology at the faculty of Technology, Policy, and Management at the Technical University in Delft.

If you are about to start reading this report I would like to thank you for your time and advise you to enjoy the ride. It has been a turbulent time for me and writing these words while the summer has been going on has been frustrating on some occasions. However, finalizing an education in Delft is a unique opportunity and I am glad to have taken it.

Of course this preface cannot do without some gratitude to all involved. First of all I’d like to thank Peter Robinnet for his time and patience, and connecting me with people involved in application development. Without his help the findings in this report could not have been made. Of course I would like to thank all the interviewees for their time and willingness to cooperate. At least as important as this thesis has been the road towards this project. I’d like to thank my fellow students for the joyful time I have experienced. It’s been a while since the start and I can assure you many things have changed. The people I have met and the friendships that have resulted are an important contribution to the period I have experienced and the life that lies ahead.

Furthermore I would like to thank my supervisory committee for their time and effort in bringing this report to an end. In particular I want to thank Erik for his invaluable support in answering numerous questions. The weekly meetings through Skype are a sign of his dedication towards his students and guided my through this project. But also Mark I would like to thank for helping me getting a foot down in a research domain and offering advice at some crucial moments. Lastly, I would like to thank my chair for asking critical questions that offered a different perspective on the project.

The Hague, August 2013
Executive summary

The world has witnessed a dramatic increase in the use of smartphones in the last 5 years. The introduction of the Apple iPhone and the Google Nexus smartphone in 2007 marked the start of a so-called ‘App economy’. It is estimated that the market for mobile application development services will reach $27bn in 2013. The market consists of different organizations ranging from application developers, to application platform providers, and companies that want to use the platform to connect to users. The success or failure of the organizations is dependent on the success of the user experience of a consumer when using the smartphone.

The role of application developers is important in this market as without them the platform will not thrive. As they are increasingly forced to collaborate with other organizations, knowledge on how they can influence their business relationships can give them a competitive advantage. More insight into how their business relationships can be influenced is expected to make them better capable in acquiring resources and appropriating on the development of applications within their business network.

The problem owner for this research are application developers. This research offers them insight into governance mechanisms that can be used to influence their business relationships in a mobile ecosystem in the form of hypotheses. The analysis has further identified practical recommendations for developers, to improve the value they extract from their relationships.

There is a mutual dependency between platform leaders and the organizations involved in the mobile ecosystem. The platform leaders need the clients to offer business opportunities to developers, and attractive applications to users. In a similar fashion, the developers need the platform leader to make a stable operating system that is bug free and offers unique possibilities for applications. The clients that want an application have to enter into a relationship with the platform leader but can only do this with use of an application developer. The mobile ecosystem can described as a multi-sided market in which the mutual dependence makes the organizations co-evolve as they work towards achieving a shared goal. A virtuous cycle is identified by the mutual dependency upon each other. It causes the ecosystem to evolve and thereby change the context of the mobile ecosystem. The market is becoming more mature, and formal control mechanisms can become more important.

Developers are good in building applications, however, they often lack business skills such as drafting contracts or commercializing their products. And although they don’t feel much competition now, this is likely to change in the future as the market matures and demand meets supply. A developer that is able to formalize his relationships can create a competitive advantage over other developers. Furthermore, it can reduce the incentive for opportunistic behaviour by an organization in a relationship. Agile contracts and subsequent project method are a method in which a good balance seems to be found in the creativity of the developer and the control of the client. They are particular well suited for projects with high technological uncertainty as is common with application development.

The current strategies of the platform leaders leave no room for any interaction. When facing difficulties, the community of application developers can offer help. There are many online communities that meet offline as well, in which resources such as knowledge is being shared. These
communities already have a large installed base, but those developers that are not actively participating are recommended to do so. Entering such a network with many different weak ties offers access to new opportunities. An environment and culture is created that fosters exploration and stimulates sharing of resources.

Furthermore, although not all developers are faced with patent infringements and licensing fees, preventing it from occurring is better than letting it happen. Staying updated with all possible patents is time consuming and unfeasible but there are databases that provide in this information and can help developers with this activity. Once a patent is still infringed and a licensing fee is requested, legal advice is necessary. Preparing for the occurrence of such an event will offer the developer a better chance of reducing any possible damage.

Lastly, there is a gap between developers and clients, a bridge player or mediator that can connect these groups with each other can appropriate on this. This can for instance be in the form of a developer that connects within the community and is able to reach outside to clients. But it can also be that relation managers (or project managers) are part of the process and form the link between the development team and the client. Clearly, this role is particular well suited for a manager in technology.
Abstract

The world has witnessed a dramatic increase in the use of smartphones in the last 5 years in particular due to the popularity of mobile applications. The market for mobile application services is part of a mobile ecosystem of organizations involved with the development of (complementary) products and services related to a platform. The mobile application developers are often in the size range of a small and medium sized enterprise (SME). Research from the perspective of an SME, in particular a developer, in a mobile ecosystem is scarce as often the focus is on platform leaders who have a strong influence in the ecosystem.

The objective of this research is to adopt the perspective of an SME in this domain and to identify if, and how, a mobile application developer can exert governance mechanisms in a mobile ecosystem to influence the relationship with other species. The results are testable hypotheses subject to future research. The unit of observation are the relationships that developers have with other species in the mobile ecosystems of Apple and Google. The unit of analysis are the governance mechanisms that can be used by the developer.

The method that is adopted is a single case study with 4 embedded cases. The embedded cases are the species with who the developer is in a business relationship; the platform leaders, clients, other developers, and patent holding companies. The newness of the research domain and the perspective of a developer, ask for an explorative research method. A case study method is chosen as it can offer a rich description of the phenomena in their real-life setting. A case study is more useful in this situation in comparison with grounded theory, as the method is open to the use of theory or conceptual categories that guide the research and analysis of data. The lack of codified knowledge on business relationships makes interviews and observations the primary sources of information for the case study. It is supported by secondary sources from websites, forums and case studies performed by other scholars. The case is analysed by pattern matching.

The findings show that exchange conditions are predictors of a governance mechanism in a transaction. The conditions are influenced by the organizations in the transaction, and the product or service that is transacted. Influencing the relationship by the developer involves safeguarding, controlling, and coordinating the economic exchange. This can, for instance, be done to create an efficient and effective business relationship, or to maximize the value gained from the transaction.

Depending on the relationship with whom the developer enters into a transaction, different governance mechanisms are more, or less, used. This has led to the development of the following hypotheses that are subject to future research.

H1: In a relationship between an application developer and a platform leader, contract-based or authority-based governance mechanisms will be used, rather than trust-based governance mechanisms.

H2: In a relationship between an application developer and a client, trust-based or contract-based governance mechanisms will be used, rather than authority-based governance mechanisms.

H3: In a relationship between an application developer and other application developers, trust-based governance mechanisms will be used, rather than contract-based or authority-based governance mechanisms.

H4: In a relationship between an application developer and a patent holding company, a contract-based governance mechanism will be used, rather than a trust-based or authority-based governance mechanism.
Content

1 Introduction .......................................................................................................................... 1
  1.1 Background ....................................................................................................................... 1
  1.2 Literature overview .......................................................................................................... 2
  1.3 Problem statement ........................................................................................................... 4
  1.4 Research objective and questions .................................................................................... 4
  1.5 Relevance .......................................................................................................................... 5
  1.6 Research approach ........................................................................................................... 6
  1.7 Report outline .................................................................................................................. 8

2 Domain description .............................................................................................................. 9
  2.1 Mobile ecosystem ........................................................................................................... 9
  2.2 Operating system ............................................................................................................. 14
  2.3 Applications ..................................................................................................................... 14
  2.4 Marketplace .................................................................................................................... 14
  2.5 Summary .......................................................................................................................... 15

3 Literature analysis .............................................................................................................. 17
  3.1 Perspectives on relationships ......................................................................................... 17
  3.2 Organizations in a mobile ecosystem .............................................................................. 21
  3.3 Governance ...................................................................................................................... 26
  3.4 Transaction cost economics ............................................................................................ 29
  3.5 Exchange conditions ....................................................................................................... 33
  3.6 Conceptual framework .................................................................................................... 36

4 Method .................................................................................................................................. 38
  4.1 Research design ............................................................................................................... 38
  4.2 Data collection .................................................................................................................. 39
  4.3 Data analysis and validity ................................................................................................. 40
  4.4 Summary .......................................................................................................................... 41

5 Case study .......................................................................................................................... 43
  5.1 Application developers and platform leaders ................................................................. 43
  5.2 Application developers and clients .................................................................................. 47
  5.3 Application developers and application developers ....................................................... 53
  5.4 Application developers and patent holding companies .............................................. 55
  5.5 Summary .......................................................................................................................... 55

6 Analysis of case study .......................................................................................................... 57
List of tables

Table 1; Distribution of SMEs among the European Union (European Commission, 2005) .................. 2
Table 2; Definition of SMEs, adapted from European Commission, 2005. ........................................ 21
Table 3; Summary of differences in characteristics between SMEs and LEs. .................................... 23
Table 4; Species in mobile ecosystem and their role in a business ecosystem................................. 25

List of figures

Figure 1; Research overview .......................................................................................................................... 8
Figure 2; Model of a mobile ecosystem (Kokabha, 2012). ........................................................................ 10
Figure 3; Conceptual framework of the relationships under research. ..................................................... 37
Figure 4; Conceptual model following the hypotheses showing directional difference hypotheses... 69
1 Introduction

In the modern world of information and communication technology, consumers require products and services that are increasingly complex and intertwined with each other, adapted to the specific needs of an end-user. Providing these offerings requires controlling the complexity of the technology and continuously adapting to the rapid advancements in the industry.

Companies need to attain innovation capabilities that do not always originate from within the organization. The high degree of interdependence in the modern economy requires small organizations to work together if they are to deliver new innovations (Ghobadian & Gallear, 1997). Participating in a network or seeing that they are part of a larger system gives them the possibility to acquire the resources needed to make them more fit to modern society. The relationships that arise from these collaborations are the topics of this study, as it are the underlying mechanisms to influence these relationships that are researched.

In this chapter the background and aim for this research are given. It starts with a description of the domain and the perspective of research that is chosen (1.1), which is followed by the theoretical background (1.2). The problem statement (1.3) is formulated, leading to a research objective and consequent questions (1.4). The relevance of this research (1.5) will be followed by the methodology (1.6). The thesis outline concludes this chapter (1.7).

1.1 Background

The world has witnessed a dramatic increase in the use of smartphones in the last 5 years. The introduction of the Apple iPhone and the Google Nexus smartphone in 2007 marked the start of a so-called ‘App economy’ (Basole & Karla, 2011; Techcrunch, 2013). Whereas the revenue from Apple’s iTunes, Software, and Services was approximately $2.4bn in 2008, this has rocketed to $13bn in 2012 and the first half of 2013 has already reported sales of $7.8bn (Financial times, 2013). Hundreds of thousands mobile software applications have since then been developed and reportedly on the platform of Apple alone 50bn application downloads have been made by May 2013 since the start in 2007 (Apple, 2013a). Mobile software applications are used on a smartphone and offer the consumer products or services. Ranging from games, to weather forecasts, and social media platforms. They are complementary to the core-product of the platform which is the operating system that is loaded on a smartphone.

It is estimated that the market for mobile application development services will reach $27bn in 2013 (ABI Research, 2013). The market consists of different organizations ranging from application developers, to application platform providers, and companies that want to use the platform to connect to users. The actual applications are downloaded (and sold) in market places or stores such as the Android Market of Google or the App Store of Apple.

In literature the business network surrounding the operating system of Google and Apple are identified as mobile ecosystems (Basole, 2009a), or more broad as software ecosystems (Jansen & Cusumano, 2012), and digital ecosystems (Selander, Henfridsson, & Svahn, 2013), to name a few. In general, it involves the business network of all organizations that interact with each other, and contribute to the development and use of the platform. Throughout this thesis the terminology of mobile ecosystems is followed which is in line with the broad definition that is given.
The core product of the platform provides an architecture for the extension with third party products by means of interfaces (den Hartigh & Tol, 2008). The complementary products make the platform attractive to the end-users, and a platform owner therefor is dependent on these developers (Gawer & Cusumano, 2008). The contributors, or developers of mobile software applications, should be encouraged to establish relationships as they drive innovation and collaboration, improving the exchanges in the ecosystem (Angeren, Blijleven, Jansen, & Brinkkemper, 2013).

The organizations in the ecosystem can be classified according to their size. A common distribution is that of small, medium, and large enterprises. In a definition of small- and medium sized enterprises (SMEs) developed in 2005 by the European Commission, their characteristics are described (among others) as having between 1 and 250 employees and a turnover between €2M and €50M. Given the broad range of this definition, it covers the majority of companies in the world and is a major contributor of innovation in the modern economy (European Commission, 2005). The distribution of SMEs within the EU show that 57,6% of the value added is derived from SMEs, providing in 67,1% of the labour force, Table 1.

Table 1: Distribution of SMEs among the European Union (European Commission, 2005).

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>SMEs</th>
<th>Micro</th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of enterprises (millions)</td>
<td>19.65</td>
<td>19.60</td>
<td>18.04</td>
<td>1.35</td>
<td>0.21</td>
<td>0.04</td>
</tr>
<tr>
<td>Share in total (%)</td>
<td>100.0</td>
<td>98.6</td>
<td>91.8</td>
<td>6.9</td>
<td>1.2</td>
<td>0.2</td>
</tr>
<tr>
<td>Number of employees (millions)</td>
<td>126.7</td>
<td>85.0</td>
<td>37.5</td>
<td>29.1</td>
<td>21.3</td>
<td>41.7</td>
</tr>
<tr>
<td>Share in total (%)</td>
<td>100.0</td>
<td>67.1</td>
<td>29.6</td>
<td>20.6</td>
<td>16.8</td>
<td>32.9</td>
</tr>
<tr>
<td>Value added (EUR billion)</td>
<td>5.360</td>
<td>3.090</td>
<td>1.120</td>
<td>1.011</td>
<td>0.94</td>
<td>2.270</td>
</tr>
<tr>
<td>Share in total (%)</td>
<td>100.0</td>
<td>57.6</td>
<td>20.9</td>
<td>18.9</td>
<td>17.8</td>
<td>42.4</td>
</tr>
<tr>
<td>Appeared labour productivity (EUR 1 000 per person employed)</td>
<td>42.3</td>
<td>36.4</td>
<td>29.9</td>
<td>38.7</td>
<td>44.8</td>
<td>54.4</td>
</tr>
<tr>
<td>Relative to total (%)</td>
<td>100.0</td>
<td>86.1</td>
<td>70.7</td>
<td>91.5</td>
<td>105.9</td>
<td>128.6</td>
</tr>
</tbody>
</table>

The size differences in organizations create differences in characteristics of the organization. SMEs often have limited resources, or resource poverty, in comparison with large enterprises (Welsh & White, 1981). Where large enterprises in turn can suffer from incumbent inertia. The differences in characteristics give SMEs both possibilities as well as limitations in their relationship with other organizations.

The novelty of the platforms make that developers are often in the size range of SMEs, although some are already large developing companies such as Rovio (Angrybirds) or Whatsapp. Throughout this research the developers that are under research are in the size range of an SME. Research from the perspective of a developer is scarce, in a research field that already has limited attention to SMEs. This study tries to contribute to this field of research by identifying what application developers, as SME, can do to influence their relationship with other organizations in their business network.

1.2 Literature overview

Mobile ecosystems can be seen as a sub-group of ‘business ecosystems’ which is a perspective on business networks that is receiving attention in literature (Iansiti & Levien, 2004; Kapoor & Lee, 2013; Moore, 1993). In a business ecosystem organizations are part of an economic community that share a common platform or service.
Research on business ecosystems makes a clear distinction between ‘central firms’ and ‘member firms’ or ‘contributors’ (Dhanaraj & Parkhe, 2006; Gulati, 1998; Kogut, 2000). The central firms, also named ‘hub firms’ or ‘platform leaders’, are defined as the centrally positioned firms in the ecosystem that have a dominant role in the platform or service (Dhanaraj & Parkhe, 2006; Gawer & Cusumano, 2008). In this sense, a platform is defined by Gawer as “a product, service, or technology that is developed by one or several firms, that serves as a foundation upon which other firms can build complementary products, services, or technologies” (Gawer, 2009). The platform leader delivers the core product and the dependency that contributors have on the platform leader gives them a power position to exert control over contributors in the process of decision making (Anggraeni, Hartigh, & Zegveld, 2007; de Reuver & Bouwman, 2012). The contributors are defined as organizations that develop and deliver complementary products and services to the core product of the platform (den Hartigh & Tol, 2008).

The organizations in the business ecosystem strive for a shared goal (Anggraeni et al., 2007; Iansiti & Levien, 2004), but to achieve that goal some form of collective action must be organized (Provan & Kenis, 2007). Organizing, or orchestrating, that collective action can be interpreted as a broad definition of governance. Although the platform leader is not the direct ‘boss’ of the other organization, they try to influence their relationship with them and steer the collaboration towards the achievement of a shared goal (den Hartigh & Tol, 2008; Provan & Kenis, 2007). Many researchers take the viewpoint from the central firm (Gawer & Cusumano, 2008; Gulati, 1998). Governing relationships in an ecosystem can give platform leaders a competitive advantage over contributors and platform leaders of competing ecosystems (de Reuver & Bouwman, 2012; Hoffmann, 2007; Ritter & Gemünden, 2004; Teece, 2007; Wareham, Lluis, & Giner, 2012). However, limited research on business ecosystems has been performed with a viewpoint from a contributor or SME.

Contributors increasingly engage in collaborations for the development of innovations (Ghobadian & Gallear, 1997). The collaborations are with other organizations that range in size but are often of the size of SME. Nevertheless, sometimes they engage in a relationship with larger enterprises that pursue a ‘keystone’ or ‘dominator’ strategy. Little is known, however, if and how SMEs can influence their relationship with these organizations, or exert governance mechanisms in these economic exchanges. Exerting governance mechanisms is for now defined as trying to influence the relationship in a business network to predict behaviour of the other organizations (Ritter, 2007). A more elaborate definition of exerting governance will be developed during the literature analysis of this research.

To analyse the relationships that organizations have in a business network, Transaction cost economics (TCE) can be used. It is seen as a first attempt in analysing the economic exchange, instead of the goods and services (Williamson, 1979). It is a first step in unravelling the complexities and with knowledge of the governance mechanisms, behaviour can be predicted in a next step with use of for instance Game theory (Axelrod & Dion, 1988; Nalebuff & Brandenburger, 1997). A different analysis on the relationships could be made with complex adaptive systems theory, which can deliver a system-level analysis of the relations in the network (Choi, Dooley, & Rungtusanatham, 2001). Also social network analysis would yield a different result as it has more focus on the structural properties of the network (Ahuja, 2000; Basole, 2009b) instead of what is happening in each relationship. And although these theories can provide in valuable insights on a network level, it are the individual
relationships of the developers that are of particular interest for this research. TCE can be used to analyse the governance mechanisms that are used in each relationship (Ritter, 2007).

Within the dimensions (also named institutions) of markets, hierarchies, and networks, the governance mechanisms can be used to influence a relationship (de Reuver & Bouwman, 2012; Jones, Hesterly, & Borgatti, 1997; Williamson, 1979). Exchange conditions of transactions raise transaction costs and can create market failure (Geyskens, Steenkamp, & Kumar, 2006). Leading to the decision to produce internal to the firm or integrate vertically. The choice of a governance mechanism is influenced by these conditions that surround a transaction. The exchange conditions are observable measures as identified by Williamson (1979) and are influenced by the organization in the transaction and by what is being exchanged.

1.3 Problem statement

Research from the perspective of an application developer has not seen much practice thus far. This is due to the novelty of the domain but also the keen interest of many scholars towards platform leaders (Gawer & Cusumano, 2008; Jansen & Cusumano, 2012; Levien & Iansiti, 2004). The market for application development is relatively young but is already growing rapidly, evolving towards a mature market, or App economy. The role of application developers is important in this market as without them the platform will not thrive (Jansen & Cusumano, 2012). As they are increasingly forced to collaborate with other organizations, knowledge on how they can influence their business relationships can give them a competitive advantage. This research aims to derive testable hypotheses of the governance mechanisms that are used in the relationships that developers have with different organizations in the mobile ecosystem. In that sense, the application developers are the problem owners.

1.4 Research objective and questions

The objective of this research is to identify if, and how, a mobile application developer can exert governance mechanisms in a mobile ecosystem to influence the relationship it has with other organizations in the mobile ecosystem. If a developer can exert governance mechanisms, research will be performed on the mechanisms that are used. If it is not possible, the barriers that prevent the developer from exerting governance mechanisms are researched.

The unit of observation is the relationship that developers have with different organizations in the mobile ecosystem of Apple and Google. The unit of analysis are the governance mechanisms that can be used by the developer. The goal is deliver testable hypotheses that are subject to future research.

To reach the objective a main research question is formulated with several sub-questions. By answering the sub-questions, an answer to the main question is developed and hence the objective can be achieved. The main research question following the objective is:

RQ: “How can a mobile application developer exert governance mechanisms in a mobile ecosystem to influence the relationship with other organizations in the ecosystem?”

The first sub-question (SQ-1) will broaden the knowledge on the concept and identify the underlying mechanisms of governance in a mobile ecosystem. It will provide in a definition of governance used throughout the research and differentiate for developers in particular. Once a definition of governance is given, research from transaction cost economics will give insight into the governance
mechanisms that can be used (SQ-2). The choice of a governance mechanism will be influenced by the exchange conditions surrounding a transaction (Williamson, 1979). How the exchange conditions influence the governance mechanisms is researched in the third sub-question (SQ-3). The answers to these sub-questions are the result of the literature analysis.

**SQ-1:** What definition of governance by an application developer in a mobile ecosystem can be derived from literature?

**SQ-2:** What governance mechanisms can be identified from literature?

**SQ-3:** How do exchange conditions influence the governance mechanisms that can be used?

It is unknown if application developers are aware of governance mechanisms and how they can be used to influence the relationships in a mobile ecosystem. However, by interviewing practitioners governance mechanisms can be identified that, perhaps unknowingly, are used (SQ-4). Lastly, depending on the specie in the ecosystem the governance mechanisms might change. A differentiation into species of organizations will offer application developers knowledge that is more applicable to their specific needs (SQ-5). The answers to these sub-questions are the result of the analysis of the case study.

**SQ-4:** What governance mechanisms can be identified in practice?

**SQ-5:** How do governance mechanisms differ according to the relationships with different types of species in the mobile ecosystem?

### 1.5 Relevance
Research on the mobile industry is receiving attention in scientific literature (Basole, 2009b; de Reuver & Bouwman, 2012; Iansiti & Levien, 2004). However, most often a viewpoint from a central firm is taken and hence there is little literature available with a viewpoint from an SME. In the following paragraphs both the relevance for managers and the scientific community respectively is discussed.

#### 1.5.1 Managerial relevance
Existing case studies (Gawer & Cusumano, 2008) have shown that platform leaders have ‘best practices’ and strategies to improve the health and performance of the mobile ecosystem and the platform leader. By exerting governing mechanisms in their ecosystem they organize the collective action to assure the shared goal is followed. However, the high degree of interdependence and rapid advancements in technology within the mobile industry require all organizations to work together if they are to deliver new innovations (Basole & Karla, 2011; Ghobadian & Gallear, 1997).

If application developers have knowledge with respect to the governance mechanisms that can be used to influence their relationships, they can predict behaviour and develop strategies for it. This can give them a competitive advantage over other organizations in the mobile ecosystem. This research will attempt to identify if such mechanisms are available to a mobile application developer and identify how this will differ for different types of species in the mobile ecosystem.

#### 1.5.2 Academic relevance
Research from the perspective of SMEs and in particular application developers in a mobile ecosystem is scarce. Most research is focused on the market places in which the applications are sold
(Eaton, Elaluf-calderwood, Sørensen, & Yoo, 2011; Jansen & Bloemendal, 2013) on the platform leaders and their strategies to govern the platform (Ghazawneh & Henfridsson, 2013; Tiwana, Konsynski, & Bush, 2010), and on the historical evolution and outlook of the platforms (Tilson, Sørensen, & Lyytinen, 2012). However, SMEs in particular have to cross through different ecosystems and the innovation landscape if they are to deliver new innovations (Selander et al., 2013). Which requires different skills and knowledge, other than is currently available for platform leaders.

Research has been performed on the factors that influence the choice that developers have in selecting a platform (Kokabha, 2012) or on their participation in the ecosystem for cultivating their innovation habitat (Selander et al., 2013). Nevertheless, the mere amount of developers and the added value to the economy of SMEs make them an important group of organizations in society.

More insight into how their business relationships can be influenced is expected to make them better capable in acquiring resources and appropriating on the development of applications within their business network. This research tries to make a first step in this direction.

### 1.6 Research approach

The nature of the study is explorative and therefore a qualitative research method will be used. Primarily consisting of an extensive literature research, interviews in a case study, and observations. A case study approach is seemed most suitable, as there is a lack of knowledge in literature. The case study approach offers the possibility to analyse and explore the underlying principles of the concept (Yin, 2011), in an early attempt to explain the governance options for an application developer.

The method is different from for instance ethnography in which the culture of the community and ecosystem could have been explored, or phenomenology in which experiences from individuals are perceived. These methods adopt a disciplinary orientation and distinguish from a functional orientation with grounded theory or a form orientation with case studies (Merriam, 2009). Also, a case study is more useful in this situation in comparison with grounded theory, as the method is open to the use of theory or conceptual categories that guide the research and analysis of data (Meyer, 2001). Whereas grounded theory will result in a theory that is grounded in data, a case study will offer a rich description of the phenomena in their real-life setting.

The analysis will be conducted at an individual actor level. Therefore the research will be a single-level analysis. The details of the method will be described in chapter 4, some essential notions are described in this section.

#### 1.6.1 Phases of research

The research will be divided in 3 distinct phases, each phase will provide in a more in-depth analysis of the topic towards the objective. The first phase will form the literature analysis of the research in which the underlying theories of mobile ecosystems and relationship governance are detailed. From these theories a conceptual framework will be developed that is applied onto the domain of a mobile ecosystem, from the perspective of an application developer. The results of the first phase are questions that can be used in the second phase to collect data from practitioners. It is in this phase where interviews will be held and observations will be made during a case study. After the interviews have been conducted, the results will be analysed. In the third and final phase the hypotheses will be developed and the conclusions are drawn to finalize the research.
1.6.2 Interview protocol
Semi-structured interviews will be conducted with application developers, and clients of application developers. An interview will give the possibility to explore outcomes that were not identified during the literature research. It is likely that the interviewee’s are unaware with the possibilities inherent to the concept of business ecosystems. To prevent bad results due the lack of knowledge of the interviewee’s the questions must be formed describing the concept. If necessary the interviewees will be informed on the topic.

The sampling method will be convenience sampling to the extent that they were involved in business relationships with other organizations. The sample preferably consists of people that have experience with managing relationships or are involved on a strategy level of the company. The interviews will be conducted at the location of the respective company or developer to provide in a natural, trusted, setting for the interviewee. The questions in the interview are developed during the literature study.

1.6.3 Research overview
The design of the research is translated into a research overview in Figure 1. It consists of the three phases as mentioned. These are the development of a conceptual framework, gathering empirical data and analysis, and developing hypotheses and conclusions.
1.7 Report outline

The following chapter will contain a description of the domain and the relevant organizations and factors that are part of it. It provides in the context for the literature analysis of the research that is in the following chapter. This will start with perspectives on relationships and networks. The aim is to get a better insight into the different perspectives and explain the implications of a mobile ecosystem perspective. The objects of research are application developers that are in the size range of SME. Their unique characteristics in comparison with large enterprises are described in this section. Governing the transaction by an SME is the subject of this research and the literature on relationship governance in the context of a mobile ecosystem is presented in the following section. It is analysed with use of transaction cost economics and is dependent on the exchange conditions. A conceptual framework concludes this chapter.

The method of the research is presented in chapter 4. Followed by the in-depth case study in chapter 5. In chapter 6 the analysis of the case study is made, guided by the conceptual framework of chapter 3. The hypotheses will be the result of this chapter. Finally, the conclusions and findings form the last chapter. The answers to the research questions are given followed with recommendations and limitations. A reflection in this chapter finalizes the thesis.
2 Domain description

The domain in which this study will be undertaken is the mobile industry. A dynamic industry due to the continuous technological advancements that has seen a rapid growth with the introduction of the Apple iPhone and the Google Nexus phone in 2007. These phones offer users a platform on which contributors can connect and offer compatible and complementary products. Since its introduction, the different platforms have acquired a market share of approximately 90% by April 2013 (BGR, 2013), and can therefore be seen as representative for the market. Other platforms are that of Blackberry’s RIM, Microsoft’s Windows Mobile, and Nokia’s Symbian.

The organizations that are involved in the development and exploitation of the platforms are members of a business network that can be viewed as mobile ecosystems (Janssen, 2009). The mobile ecosystem can cross different industries and networks but connects the different organizations, networks, and industries with each other through the platform. The ecosystems of Apple and Google are the domain of this research.

Within the ecosystems of Apple and Google, it are the application developers that form the focus of this research. As mentioned, research from the perspective of an application developer has not seen much practice thus far. This is due to the novelty of the domain but also the keen interest of many scholars towards platform leaders (Gawer & Cusumano, 2008; Jansen & Cusumano, 2012; Levien & Iansiti, 2004). The role of application developers is important in the App economy as without them the platform will not thrive (Jansen & Cusumano, 2012). As they are increasingly forced to collaborate with other organizations, knowledge on how they can influence their business relationships can give them a competitive advantage.

In this chapter a description of mobile ecosystems is given and the concepts that are related to in particular the platforms of Google and Apple. In the following section the organizations that are part of the ecosystem are described. In involves application developers as they form the perspective that this study takes. The next section discusses the operating system and its role in the ecosystem. The complementary products, or applications, are discussed in the following section. They have contributed to the success of the current platforms and are expected to be of great importance in the near future. The applications are sold in markets and stores which are described in the next chapter. A summary finalizes the chapter.

2.1 Mobile ecosystem

Previous research in the mobile industry has often taken a perspective of a value chain (Barnes, 2002; Karvonen, 2004). However, the growing interconnectedness of the organizations and the evolving nature of the platform, limits the use of this perspective. The large variety of firms from different segments meet in a business network that has a good fit with a business ecosystem perspective. More recent research has taken this perspective (Basole, 2009a), and in particular a mobile ecosystem perspective as sub-group of business ecosystem (Angeren et al., 2013). The mobile ecosystem shares the properties of a business ecosystem but is more specific to the context in which it is situated.

The business network contributes to the platform’s survival by reinforcing network effects to grow the installed base, and sharing and aligning resources to improve the technology (Selander et al., 2013). It is a co-evolutionary effect between participants and the technology that creates self-
reinforcing feedback loops. These can make participants both collaborators as well as competitors as it can have both positive and negative effects (Moore, 1993; Selander et al., 2013).

The platform central to the mobile ecosystem is based on an operating system on which a wide variety of applications can be downloaded and installed by users. It involves a common set of components and rules that is used to interact with each other (Eisenmann, Parker, & Alstyne, 2011). With the applications they can use their smartphone as game player, navigation aid, remote control, and many other utilities. It has led to a fierce battle between platform leaders to offer the most interesting platform to its users and organizations in the ecosystem (Basole & Karla, 2011). The current leading platforms are iOS and Android, of Apple and Google respectively. The enormous market opportunities have led to the entry of many new participants (Basole, 2009a). And has changed the environment for many existing organizations. Where network operators used to have the most control in the network, the platform operators seem to have control now. The mobile ecosystem can schematically be visualized according to Figure 2, (Kokabha, 2012). The results of the research from Kokabha involve an adapted schematic view from the concept Moore proposed in 1996, Figure 2. The relevant organizations that participate in the mobile ecosystem are described in the following paragraphs.

![Figure 2; Model of a mobile ecosystem (Kokabha, 2012).](image)

### 2.1.1 Platform leaders

The platform leaders of the mobile ecosystems under research are Apple and Google. Where iOS is introduced in the market by June 2007, Android followed in November 2007. They offered the users a new experience by which they could download and run applications, changing the smartphone into a multi-functional utility device. Although both ecosystems quickly saw a sharp growth in adoption, there are large differences in the strategies that both platform leaders followed.

Apple controls the entire ecosystem by itself, thereby safeguarding the quality level of the look and feel that users experienced (Ghazawneh & Henfridsson, 2013). It can be described as the ‘Apple
experience’ in which all applications have similar design features to the operating system and the applications are almost completely bug-free\(^1\). Apple achieved this by developing guidelines to which applications developers should adhere, and developed as much as possible of technology of the core product in-house, including actual devices. The applications are sold in the App Store, the only portal for users of iOS to acquire applications for their phone.

In contrast with Google, who has allied with numerous partners in the development of the Android platform (Kokabha, 2012; Open Handset Alliance, 2008). The operating system is developed by a company that was acquired by Google in 2005. Together with 34 other organizations they started the Open Handset Alliance (OHA) (Open Handset Alliance, 2008), for the development of a smartphone and operating system that focused on the use of open standards to reduce costs for the entire industry. It led to a sharp growth in users and organizations that participated in the ecosystem. Although the operating system is of high quality, it was not bug free but left the user many control over how the device should be used (Kokabha, 2012). The applications were sold in the Android Market, or Google Play, however, quickly many other marketplaces followed as it is a user free to choose were to purchase or download applications. Nevertheless, the ‘Android experience’ suffers from the lack of coherence and quality as there is limited control on the applications and device manufacturers that load there products with Android (Kokabha, 2012). Furthermore, as most applications are free, it is difficult for developers to appropriate on the platform which limits the effort and time that is spend on testing to prevent bugs. As a result, the overall quality of the Google platform suffers.

From the business ecosystem analogy, platform leaders can be seen as an organization that follows a ‘keystone’ or ‘dominator’ strategy. Keystone players (catalysts or shapers) follow a strategy in which they try to improve the overall health of the business ecosystem. By stimulating exploration of new business opportunities and enabling continuous renewal of the partner’s ecosystem, they increase the health of the ecosystem. They offer a stable and predictable set of assets that other organizations can use to build their own offerings (Iansiti & Levien, 2004). The self-interests of the partners allows for new business developments to appear from the partners and members instead of being organized by the platform leader.

With a leading role in the development of innovations within the ecosystem they give direction towards the shared goal. By branding they link customers to suppliers and stimulate the creation of a community (Stuart, 2000). They establish relationships with members of the ecosystem based on trust through endorsement, certification, and references (Stuart, 2000).

Whereas keystone players exercise power by helping others, there are players who exercise power to the benefit of their own. Dominators try to exploit the ecosystem by taking over a large portion of the network as physical dominators, or by extracting value from the ecosystem as value dominators (Iansiti & Levien, 2004). The dominator will try to orchestrate or organize the network by direct account manager contact with the partners. This can be reached more effectively by developing partner programs in which partners are obliged to conform to certifications. With the entry to this

---

\(^1\) A bug is a malfunction in a program which can temporarily cause the application to halt operation. Bugs are considered annoying and the applications that are bug free are considered of higher quality than applications that suffer from bugs.
program they are obliged to give insight into the business opportunities and steer the partner towards the outcome the dominator prefers (Hartigh, Visscher, Tol, & Salas, 2011).

It can be a successful strategy to pursue but will become less flexible and will provide in less innovations in comparison with a more open ecosystem (Iansiti & Levien, 2004). It can become vulnerable to external forces in the long-term and be surpassed if it does not carefully assess its environment and competitive position.

2.1.2 Application developers
The application developers in the ecosystem are involved with the development and distribution of mobile software applications. Since the introduction of the platform, many developers have entered the market due to the large demand. Some have become large enterprises with the success of mobile applications, such as ‘Angrybirds’ or ‘Wordfeud’. Others remain a self-employed developer.

The applications are developed with different coding languages and the ability to produce an application for both ecosystems requires the developer to ‘code’ in both languages. However, with the possibility to develop applications on both platforms, the developer can reach 90% of the market. Many developers enter the market with the development of just one application and most often stay with just one application. As a result, 71% of the developers have just one application, whereas 5% has more than 16 (Visual, 2013). Many start-ups have merely one application and there are little larger developing companies. Nevertheless, the developing companies that are able to produce several successful applications often have rapid growth rates.

The developers meet each other both offline as well as online. Meeting offline, in the Netherlands, involves participating in social events, such as Appsterdam, or in more knowledge focused or ‘Tech’ events such as Cocoaheads. The aim of these events is to broaden the social network of developers and share their knowledge with others. This helps developers in staying updated with the rapid advancements in the technology. Online they meet at forums such as stackoverflow.com, crunchbase.com, and social media platforms such as twitter. They help each other with solving problems ranging from technical issues, to more business related issues.

From the revenue that is created with the sales of the application 30% is given to the platform leader (in the case of Apple or Google) (Apple, 2013b; Google Play, 2013). The revenue can be created from the actual sales of the application but also in sales of functionality within the application, or advertisement. Furthermore, the developers produce applications for clients. In a study by Interbrand 2012, Best Global Brands, 90 out of the top 100 brands now have an application in at least some application store or market (Interbrand, 2012).

The developers can work for clients such as brands that need complementary products for their core business. In other occasions, the developer works for himself, in an attempt to gain revenue from the sales of the application in a marketplace.

2.1.3 Clients of application developers
The market for application developers is generated by the demand for applications by clients. A client can be any organization (or individual) that wants an application developed. They can range in many different sizes, from large brands such as Electronic Arts, to self-employed individuals that want an application developed.
The need for an application can be generated as the organization has an idea that it wants developed. The idea can be a new primary product for an organization, or it can be complementary to existing products or services. It can be produced for a platform and released in the marketplace. Or it can be developed as application internal to the company that is not made publicly available. The demand for application development is growing rapidly as organizations realize they can reach a worldwide market through an application (ABI Research, 2013).

2.1.4 Device manufacturers
The device manufacturers are involved with the development of the actual smartphone. Within the platform of Apple the development is in close collaboration with Apple. The entire supply network is coordinated and controlled by Apple. In contrast with Google, which allows device manufacturers to develop their own smartphone and load it with an Android operating system free of charge. As a result, there are approximately 300 different smartphone and tablet devices that run Android, in comparison with 10 smartphones and tablets in the Apple ecosystem.

A proper functioning of the smartphone requires the alignment of both hard- and software. As there are so many different devices with Android, the alignment is more difficult and has made smartphones loaded with Android more bug sensitive than Apple. In particular application developers experience difficulties in addressing the functionalities of the hardware needed in support of their application.

2.1.5 Network operators – Internet service providers
The role of network operators has changed dramatically over the past years. Where they controlled large parts of the network in the past, they seem to have limited control today (Basole & Karla, 2011). Nevertheless, they provide in the infrastructure and connection to the network. Often this involves large investments in technology to keep up with the rapid advancements and the increasing demand of consumers. The business models of the operators are changing as consumers are increasingly using the internet instead of calling or texting SMS. They are now focusing on offering high speed mobile internet connections, to allow users to maximize their experience on their smartphones. With this approach, they are stepping away from the previous pay-per-minute model in which the operator had strict control over the usage of the phone, as calling was the primary functionality. With the availability of internet on smartphones, operators can no longer control what functions are used and charge accordingly.

2.1.6 Patent holding companies
The mobile ecosystem is accompanied with rapid advancements in technology. Many parts of these technologies are patented to allow appropriating on the technology as it is incorporated in different devices. It has led to a group of companies that merely own patents and require licensing fees from organizations that use the patents. They assess if use of a technology is violating a patent they own and file law suits to the organization that is using it.

Similar activities are present among platform leaders as they own many patents for their platform. It has led to fierce patent battles between platforms, hindering innovations from others and preventing sales in certain areas (Puttk, 2013).
2.2 Operating system
The operating system (OS) can be seen as the core product of the platform that is loaded on the smartphone. Users join the platform by buying a smartphone that is loaded with an operating system of the platform leader. With the operating system, platform leaders try to attract more developers and consumers to their platforms.

The application developers program their applications in language that is used by the operating system. For iOS this is Objective C (Iphoneclub, 2013), whereas on Android this can be different but is predominantly Java (AndroidWorld, 2013). The operating system connects with the hardware to offer the user control of the hardware functionality on the smartphone. Aligning soft- and hardware involves controlling the interfaces of the different components of the smartphone and making this accessible for the user or developer.

The Android operating system is an open source platform but is controlled by Google in the end. Developers can contribute to functionality of the system in the Android community. Functions can then be incorporated in an update of the OS, if Google decides to use the functionality. Thereby Google controls a part of the system to safeguard the quality. The open-source nature of the operating system makes it an open platform for organizations to contribute, but purely for the development of the platform.

iOS is developed by Apple and keeps the development in-house. It is a closed platform in which Apple has strict control over the look and feel of the system (Apple, 2013c).

2.3 Applications
The applications that are being developed by the organizations in the ecosystem offer the users a broad range of functionalities. This can range from social media, to news services, and games, they are dependent on the platform on which the application is published as that determines the programming language of the application. Depending on the platform that is chosen, a similar but different market is addressed as users often don’t have both an Apple and Android device. Organizations, both developers and brands, can choose to develop applications that are available on both platforms. Nevertheless, it requires the development of similar but different applications.

The applications can see different types of usages. Applications can become a hype for a short period of time but have little to none recurring users, as their content is not updated. The developers or brand will gain revenue from the sales of the application but will quickly loose the installed base that has grown in the short period that the hype lasted. Other applications are continuously updated and make it more attractive for a user to use the application more frequently. These can be seen as utility applications (Pagkozidis, 2012).

With the platform of Apple, new applications are subjected to a quality screening and if the application does not meet the requirements, it will not be allowed in the App store in which the applications are sold (Apple, 2013c). Applications on the Android market are prone to a short screening and can be submitted almost directly (Google, 2013).

2.4 Marketplace
The applications are sold and distributed via an online store or market. The application aggregators offer developers a portal by which they can reach a worldwide market of users. Users, in return, can
search the market or store for the application they need. For the Apple platform, the App store is the only channel by which applications can be acquired. With Google there are many different portals, although the Google Play, or Android market, is the most popular one.

The application store can be defined as “an online curated marketplace that allows developers to sell and distribute their products to actors within one or more multi-sided software platform ecosystem” (Jansen & Bloemendal, 2013, p.195). It is the channel through which brands and companies can connect to their customers or end-users for the distribution of applications. Within the store or market, there can be policies with respect to for instance use of an application or admission to the market. But also features such as the possibility to search for applications and marketing or advertisement of applications.

The stores and markets are part of the platform of the ecosystem and can have a pivotal role in creating a market (Jansen & Bloemendal, 2013). They make access to customers easier than through communication channels of the individual brand or developer. Furthermore, they reduce distribution costs as the portal is central to many users and developers (Jansen & Bloemendal, 2013).

Revenue for the platform leaders and owners of the application is created through the application market or store. This can be through purchase of the application, or via in-App purchases, e.g. purchases that are made with-in the application to gain for instance extra content. Also, revenue can be created through advertisement within the application. In all cases, a part of the revenue goes to the platform leader and the other part to the owner of the application (Kokabha, 2012).

2.5 Summary
The organizations that are involved in the development and exploitation of the platforms are members of a business network that can be viewed as mobile ecosystems (Jansen, Brinkkemper, & Finkelstein, 2009). The mobile ecosystem can cross different industries and networks but connects the different organizations, networks, and industries with each other through the platform. The ecosystems of Apple and Google are the domain of this research. A mobile ecosystem perspective is better in grasping the co-evolutionary aspects of the platforms in comparison with more traditional methods such as value chains.

The core product of the platform is the operating system that is developed or controlled by the platform leader. It is loaded on a smartphone and by buying a smartphone a user connects to a platform. The application developers contribute to the ecosystem by developing complementary applications that increase the functionality of the smartphone. Users can use their smartphone for different functions ranging from navigation device, or newsreader, to email client, and camera. Apple develops both hardware (the iPhone) and software (iOS) in-house and has a closed platform. Google, in contrast, allows device manufacturers to load their smartphones with Android free of charge. Furthermore, the operating system Android is an open-source project, although Google remains in control over the actual changes.

The developers build applications for their selves to distribute in the marketplaces, or for clients that want to make use of the business opportunities in the mobile ecosystem. The clients can be brands that offer extra service to their main product or companies that have an application as core product. The applications are sold in an online marketplace such as Google Play, for Android/Google, or the App Store, for Apple. The revenue that is created with the sales of applications or from usage of an
application is shared with the platform leader. In the case of Apple and Google, they both require 30% of the revenue. The marketplaces offer an easy connection between developer/client and a user on a worldwide market, from a central place. This allows the distribution costs to reduce and makes it attractive to users to connect and use that market place.
3 Literature analysis

The mechanisms by which SMEs and in particular application developers can exert governance in a mobile ecosystem are the topic of this research. In this chapter the underlying principles and theories are discussed that form the theoretical framework.

The first section gives an overview of different perspectives on relationships. New developments in science have led to perspectives that are increasingly better able in explaining relationships between organizations and how resources are exchanged. The summarizing section elaborates why the perspective of business ecosystems is more complete than others in explaining economic relationships in society. The mobile ecosystem perspective as sub-group of business ecosystems is used throughout this research.

The following and second section discusses the role of SMEs and the specific characteristics that make them distinct from large enterprises. Their characteristics give both possibilities and limitations during economic transactions with other actors, or organizations, in the ecosystem. From the perspective of a business ecosystem the other organizations can be classified in different roles. As in natural ecosystems, certain roles are essential for a healthy ecosystem. They help other organizations or are of influence in the ecosystem in realizing a shared goal. This section concludes with a summary of characteristics of both SMEs and the roles in the context of the ecosystem.

The third section discusses the topic of governance as described in literature and the concept that is used in this research. It starts with an overview of the different interpretations of governance that is found in literature. Throughout this research the concept of relationship governance in the context of a mobile ecosystem is used. It is distinct from corporate and public governance and can be seen as a sub-group of network governance. In particular it are the relationships that are governed in the mobile ecosystem.

Transaction cost economics is used and elaborated upon in the fourth section to be able to analyse the relationships. The different dimensions as proposed by Williamson (1979) are discussed as well as the criticism that has led to additions in recent literature (Jones et al., 1997; Powell, 1990). The governance mechanisms that can be used are dependent on the exchange conditions and the organizations that exert them. They are discussed in the fifth section. The exchange conditions indicate when a dimension is more effective or efficient than another. The chapter concludes with an overview of this chapter and shows how the different concepts are related with each other in a conceptual framework.

3.1 Perspectives on relationships

In an attempt to unravel the complexities of economic reality, different perspectives on relationships between companies, industries, and industry crossing networks have been developed. This chapter discusses several perspectives that can be used to look at companies and their networks. In the final paragraph the choice for a mobile ecosystem as sub-group of a business ecosystem is elaborated upon.

3.1.1. Dyadic

Research on dyadic relationships is focused on “the characteristics and attributes of organizations to explain their relationship with other organizations” (Provan & Kenis, 2007, p.3). The involvement with other organizations is through different types of dyadic relationships such as joint ventures,
alliances, and partnerships. It is used to understand the nature of the relationship between organizations in terms of their characteristics and attributes (Ahuja, Soda, & Zaheer, 2011; Provan & Kenis, 2007). These characteristics incorporate, among other, tie strength, or the degree of trust, and how these relational characteristics affect the likelihood of the relationship’s renewal, continuation, dissolution, or other outcomes (Ahuja et al., 2011).

Current research is on a broad range of topics such as trust, ties, and embeddedness. The level of trust between two organizations can be improved if the organizations repeatedly enter into ties with each other (Gulati, 1998). Furthermore, the strength of the tie can foster specific types of asset sharing, whereas strong ties are better for the transfer of tacit knowledge, weak ties foster the transfer of explicit knowledge (Uzzi, 1997). Ties that have embedded (social) relationships between managers of different organizations are said to improve firm performance (Ingram & Roberts, 2000). Whereas “over-embeddedness can hurt performance due to the limited diversity of information to which they have access” (Ahuja et al., 2011, p.66). With respect to trust, Zaheer (2011) has shown that high levels of (inter-organizational) trust between organizations can reduce transaction costs and “allow for the extraction of higher benefits from the relationship” (p.66).

The dyadic perspective gives a view on the impact of a relationship between organizations but with assumption of ceteris paribus. Thereby it is ignoring, among other things, that organizations and a relationship between organizations are likely to be influenced by the overall set of relationships in an industry or network (Provan & Kenis, 2007). As a result, a network that is researched from a dyadic perspective consists of a collection of two-party relationships, rather than a unique, multi-organizational social structure (Provan & Kenis, 2007).

3.1.2 Ego-network

The ego-network perspective is focused on the structure of relationships surrounding the focal organization (ego). It refers to the kind of effects that an ego’s network has on its behaviour and performance. It can help analyse the impact of individual organizations, and dyadic or network ties, on organizational performance (Ahuja et al., 2011; Provan & Kenis, 2007). And is focused on the role of an individual in a network. Key issues relate to understanding the structure of relationships with structural issues such as: centrality, structural holes (and closure), structural embeddedness, structural equivalence, social capital and lastly status (Ahuja et al., 2011).

The structural properties, such as centrality and the size of the alliance network, of an ego’s network have been a field of interest for many researchers in the past decades. The properties have been connected to different meanings of performance such as improvements in absorptive capacity (George, Zahra, Wheatley, & Khan, 2001), innovative output (Ahuja, 2000), the rate of new product development (Deeds & Hill, 1996), and increased venture capital investments (Baum, Calabrese, & Silverman, 2000). Another field of interest is that of social capital. Networks with many structural holes are rich in social capital (Burt, 1997), whereas (Coleman, 1988) has argued that closure does. Reconciling these conflicting views, (Burt, 1997) makes a distinction for different situations in which either structural holes or closure improve social capital. In situations where there is cooperation, closure in the network will improve social capital. In situations where there is fierce competition, structural holes will improve social capital. A last field of interest is that of ‘status’, as being a signal of quality and aiding in finding partners for exploring new domains (Dimov & Milanov, 2010). A
higher status compared to competitors can lead to lower transaction costs in acquiring resources (Podolny & Page, 1998).

Research on networks generally has two perspectives in literature, the view from the individual organization with regard to its network, and the overall view from the network level. In literature, this distinction is also seen as the difference between a micro and a macro level analysis (Wasserman & Faust, 1994). A distinction is made between egocentric network versus whole network respectively (Ibarra, Kilduff, & Tsai, 2005). An important limitation of this perspective is that it focuses on a single industry, limiting the generalizability of the findings.

3.1.3 Whole network
The characteristics and behaviour of the entire inter-organizational network (Ahuja et al., 2011) are the focus of a whole network perspective. It involves analysing the impact of multi-level actions and structures on network level outcomes or a system-level approach (Provan & Kenis, 2007). Structural concepts such as centralization and “small worldness” are key issues, together with the analysis at the whole network level instead of the individual level (Ahuja et al., 2011; Provan & Kenis, 2007). Recent attempts are made to discover network dynamics (Ahuja et al., 2011).

In a study from Venkatraman & Lee they show the evolving of networks with characteristics on both a micro and macro level of analysis (Lee & Venkatraman, 2006). The authors suggest the need for research that helps understand the dynamics of competition and cooperation in a network. This is supported by Powell et al., who show that in time collaborations are often cross-cutting, meaning that an actor can change from collaborator to competitor over time, showing how a network evolves (Powell, White, Koput, Smith, & Owen-smith, 2005). Ahuja et al. (2011), argue to perform research into ‘network dynamics’, in a broader sense. They argue that without knowledge of the genesis and evolution of the network structures, the outcome of the network is incomplete and potentially flawed. They develop a framework for research into network dynamics to help understand “how and why organizational networks emerge, evolve, and change” (p.434). They define the subject area and identify key dimensions on which networks can change.

The whole network perspective has a focus on a single industry, thereby limiting the generalizability of the findings. Furthermore, it lacks understanding of dynamics in networks. As with most network analyses, researchers face practical difficulties in obtaining longitudinal network data (Ahuja et al., 2011).

3.1.4 Business Ecosystems
The focus of research in business ecosystems is on relationships, interactions, and dynamics at the system level (Anggraeni et al., 2007). It is a view on business networks and the relationships and mechanisms that shape it. It incorporates both an ego-network approach, with the roles and strategies of the individual actors, as a whole network approach, how the network coevolves towards a shared goal. This shared goal is seen as the forward-looking vision of the industry leaders (Moore, 2005).

It is a perspective in trying to understand “the relationships or interactions among the members and their environment, the roles and interests of the members of the system, and the mechanisms guiding these interactions toward the achievement of a shared goal” (Anggraeni et al., 2007, p.11).
The key issues are in understanding the complex inter-firm relationships between actors. As well as the keystone species that have a strong influence on the coevolving dynamics in the business network (Moore, 2005). The research is divided in the characteristics and roles of firms, and the structure and dynamics of networks. The dynamics in the network influence the governance choice and performance of actors in the system (Anggraeni et al., 2007). And creates a mutual dependence among each other in which multi-sided market effects can be identified (Eisenmann, Parker, & Alstyne, 2006). The dependency upon each other makes them more willing to cooperate as they can meet again in the future.

Issues such as trust and evolution can be identified that may be unique to a particular level of network analysis (Provan & Kenis, 2007). Demarcating a distinction between dyads, whole networks, and business ecosystems. As suggested by Moore (1993), the system is more than a network, as it incorporates governmental bodies, associations, standardization bodies, and crosses different industries and networks. It incorporates both relational embeddedness (as concept from the dyadic level) and structural embeddedness (as concept from the ego-network level). The perspective has the ability to open the black box of the co-evolution of outcomes, behaviours, and structures (Moore, 2005).

In the ecosystem companies are defined by its innovation trajectory instead of their products and service offering. Every company is forced to constantly update their products and services in an attempt to keep up with competition. They co-evolve with the other actors in the ecosystem as they are mutual dependent. This view is different from the traditional view of the firm, which incorporates incremental innovations to existing products instead of aggressively looking for new possibilities and solutions in a more radical way to keep a competitive advantage over the other actors in the ecosystem (Moore, 2005).

The perspective of business ecosystems draws on analogies from natural ecosystems. Although there is a good fit with concepts between a business ecosystem and a natural ecosystem, there are differences. The key differences are a conscious choice, competition, and the ability to produce innovations (Peltoniemi & Vuori, 2004). Hence, there is a point at which the metaphor does not hold. It can be argued that “complexity in social systems should be studied on their own right and not as analogies from natural sciences” (Peltoniemi & Vuori, 2004). Furthermore, the business ecosystem perspective is a broad view that incorporates many still unknown complexities. New research is being performed in this field in an attempt to unravel the complexities (Basole, 2009a; Kapoor & Lee, 2013; Teece, 2007).

### 3.1.5 Summary

A dyadic view on relationships within a business network is incomplete from reality. In keeping everything but the relationship with another organization constant, it does not keep in mind the consequences that a decision can have on other organizations. Although a network perspective offers a more complete view on economic institutions, it is unable to grasp the dynamics within the network. Furthermore, the boundaries of one network limit the completeness of the perspective.

The structural properties of network analyses that have been topic of recent research are analysed by making use of graph theory, or social network analysis. However, these theories have difficulties in grasping the content of relations, and the dynamics within a network, which speaks for a different, softer approach, to network analyses.
The business ecosystem perspective allows for a view on different networks that are interrelated with each other, surrounding a central platform or service offering. Furthermore, it is better in grasping the dynamics within the ecosystem as it is co-evolving towards a shared goal. Lastly, the perspective is more suitable in relationship to whole-network or ego-network level of analyses in explaining how actors can be both competitors as well as collaborators in time. Throughout this research the business ecosystem perspective is used to analyse the underlying mechanisms of governance that companies can use in a business network. It involves both the individual relationships that developers can have with other organizations, as well as a system level perspective on the co-evolving nature of the business network.

3.2 Organizations in a mobile ecosystem

From the perspective of a mobile application developer the research is conducted. The developer has relationships with other organizations in the business network, that have been identified in chapter 2. These other organizations can be classified by their size, and in different species from business ecosystem theory, adapted from natural ecosystems. The relative newness of the market for application developers make that they are often in the size range of SME.

In this chapter the characteristics of SMEs are discussed as well as the species in a business ecosystem. In the following section the differences in characteristics of SMEs in relationship to large enterprises are given. These distinct characteristics give them both possibilities as well as limitations in doing business with other organizations. The possibilities and limitations will effectuate depending on the different roles of the other organizations in the transaction. The species and the effect of variety among them are discussed in the subsequent paragraphs. A conclusion is drawn in the final paragraph.

3.2.1 Small and Medium sized Enterprises

Different definitions exist on the size range of small and medium-sized enterprises. Commonly used is the definition by the European Commission, others are that of Eurostat, and the OECD. Outside the EU the definitions are similar. In a study by (Hauser, 2005), it is argued that a strict limit in turnover, or employees, will not define small or medium sized enterprises. The qualitative characteristics of the organization make an organization small or medium sized. However, assessing the qualitative characteristics of each individual organization is time consuming and not feasible once applied on a larger scale. Hence, as common in literature and for the sake of clarity, throughout this research the definition developed by the European Commission (2005) is used, Table 2.

Table 2: Definition of SMEs, adapted from European Commission, 2005.

<table>
<thead>
<tr>
<th>Category</th>
<th>Micro</th>
<th>Small</th>
<th>Medium</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTE</td>
<td>&lt;10</td>
<td>&lt;50</td>
<td>&lt;250</td>
</tr>
<tr>
<td>Annual turnover</td>
<td>&lt;€2M</td>
<td>&lt;€10M</td>
<td>&lt;€50M</td>
</tr>
<tr>
<td>Annual Balance</td>
<td>&lt;€2M</td>
<td>&lt;€10M</td>
<td>&lt;€43M</td>
</tr>
</tbody>
</table>

These limitations in size give the different types of organizations distinct characteristics. As stated by (Welsh & White, 1981), “a small business is not a ‘little’ big business”; differences exist in structure, policy making procedures, and resources (Ghobadian & Gallear, 1997).

Depending on the type of environment a company is faced, different organizational forms are more appropriate. The organizational forms of large enterprises are usually bureaucratic, relying on the
formalization of behaviour to achieve coordination (Ghobadian & Gallear, 1997). The organizational form of SMEs is more likely to be organic, consisting of flexible and informal working relationships. In dynamic high-tech environments in which the key to survival is innovation, an organic structure is better suited. Whereas in stable environments, formalized behaviour of a bureaucratic organizational structure is favoured (Ghobadian & Gallear, 1997).

When looking at the management of large organizations, they are most often divided in several layers. The division of functions, labour, and the span of control considerations result in the creation of a hierarchy of authority (Ghobadian & Gallear, 1997). This makes top managers far removed from operations, and hence they lack knowledge of operational issues such as quality level, customer demands, and supply chain difficulties. The various management layers will slow down the speed of communication and lead to fragmentised decision making. The centralized position of managers within SMEs makes them able to show leadership by example and catalyse or completely block decisions (Ghobadian & Gallear, 1997). Furthermore, the flat structure, with fewer interfaces, makes work more flexible and offers management the possibility to build stronger personal relationships with employees. However, it must be noted that in turn this can lead to personal conflicts (Ghobadian & Gallear, 1997). Communicating and working within an SME seems to be based more on interpersonal trust between managers and employees than within large organizations, which is more based on formalized contracts and norms.

The resources available to an SME are most often limited in comparison to large enterprises. In contrast to the stakes involved for the SME, which are most often high, relative to a similar involvement of a large enterprise. This asks for a deliberate choice in which the SME is aware of its environment and the consequences of the outcome. This is referred to as ‘resource poverty’, and requires a distinct management approach (Welsh & White, 1981). For a start, the salary of managers takes up a large part of the revenue within small organizations. This limits the organization in acquiring operational services such as accounting, but also in employee training and facilities. Furthermore, external forces such as changes in government regulations, tax laws, and interest rates can have more impact on smaller organizations in which the fraction that is influenced compared to the total revenue is larger (Welsh & White, 1981). SMEs most often lack the resources to survive such forces and remain profitable in the long run.

Cultures within an SME are likely to be different from that in large enterprises. This is, among others, due to different backgrounds of employees, staff turnover, the age of the organization, and geographical dispersion (Brunetto & Farr-wharton, 2007). Although cultures are informal and perceptual, they usually dictate what activities or behaviours are necessary to become successful within an organization. Changing a culture of an organization is thought of as more difficult in a large enterprise then it is in an SME. This is largely due to a resistance to change that is characteristic to most large organizations.

However, this can also be visible within smaller organizations, as for instance managers tend to resist documenting their understanding. They consider documenting to be a waste of time but it will also make them less indispensable as tacit knowledge is being codified (Ghobadian & Gallear, 1997). In another aspect, SMEs, that have recently invested in developing rules and procedures, might be reluctant to change again as it will make the previous investment obsolete.
When looking at the individual level, managers in an SME have different risk profiles than managers in a large enterprise. In SMEs where the owners’ personal wealth is tied up with the firm, the entrepreneur tends to be risk averse. Whereas in start-ups that have received investments from venture capital organizations or the like, the entrepreneur might be tempted to take on more risks as a part of the risk is carried by the (external) investor. Looking at large enterprises, the managers most often take on more risk, as they do not bear the risk themselves. Furthermore, as the SME entrepreneur has limited time to screen the information that is provided, he is more prone to adopt trust-based decision heuristics (Pittino & Mazzurana, 2013).

The characteristics of SMEs in comparison with large enterprises can be summarized in Table 3. Most prominent is resource poverty that is characteristic to SMEs. This limits the SME in their ability to innovate and creates a dependency on other organizations in the business network. Furthermore, the high-interpersonal trust between managers and employees stimulates the creation of more informal relations which can manifest in a more informal organization. This is supported by the organic organization structure in which people are more equal to each other and in which power distances are less large. In stark contrast with large enterprises in which the bureaucratic organization structure creates power distances and low-interpersonal trust between people involved in the organization. The directive leadership is necessary to increase the speed of decision making which is by nature slow due to the many people involved.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Small and medium sized enterprises</th>
<th>Large enterprises</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Management</strong></td>
<td>Leadership by example</td>
<td>Directive leadership</td>
</tr>
<tr>
<td><strong>People</strong></td>
<td>High interpersonal trust</td>
<td>Low interpersonal trust</td>
</tr>
<tr>
<td><strong>Policy making</strong></td>
<td>Few decision makers</td>
<td>Many decision makers</td>
</tr>
<tr>
<td><strong>Processes</strong></td>
<td>Negligible resistance to change</td>
<td>Incumbent inertia</td>
</tr>
<tr>
<td><strong>Resources</strong></td>
<td>Resource poverty</td>
<td>Resource wealth</td>
</tr>
<tr>
<td><strong>Structure</strong></td>
<td>Organic organization structure</td>
<td>Bureaucratic organization structure</td>
</tr>
</tbody>
</table>

### 3.2.2 Species in business ecosystem

The ideal business ecosystem by (Moore, 2005) has strong leaders who co-envision and co-manage co-evolution among the members. The leaders are the actors who govern the community to achieve collective action. The member firms co-create their future and should have focus on working together to improve their capabilities making them able to enlarge the market. There are different prominent species within the business ecosystem. As in natural ecosystems, Iansiti & Levien (2004) have suggested the species of keystones, dominators (or landlords), and niche players. The majority of the organizations in the ecosystem, however, specialize in a certain product or service to develop compatible and complementary products (den Hartigh & Tol, 2008), they can be seen as the contributors to the ecosystem or complementors.

An organization can evolve as specie of a keystone, dominator, or niche player irrespective of their maturity and size. It is a result of their past contributions to the ecosystem and how this is experienced by the other members of the ecosystem. It can be argued that certain species are better
suited to specifically sized companies. For instance, keystones need resources to allow other companies to build upon the assets they provide (Moore, 2005). This makes keystone players most often large enterprises.

However, the specie of organizations in an ecosystem is not static. As niche players evolve they can become keystone organizations of their own ecosystem or take over the role of the current keystone organization. Furthermore, they can participate in different ecosystems, being a niche player in a certain ecosystem and a keystone player in another.

The mobile ecosystem has many niches of which application development is merely one of them. A niche player tries to develop unique capabilities that make it ‘special’ in a particular niche. This can be as specialist or expert in a part of the central product offering or by trying to become market leader in a certain role or modular offering that is part of the larger ecosystem. They leverage complementary resources from other niches or keystones for the improvement of the central offering (Iansiti & Levien, 2004). Innovation is critical to their success as they are often faced with fierce competition from other niche players but also keystones and dominators. If they do not stay ahead of competition, or at least keep up with them, they face being taken over by the keystone organization. Differentiation can prevent this from occurring, by making the niche player less dependent on the keystone players. This indirectly forces the keystone to be honest and prevent them from becoming a dominator by extracting too much value from the ecosystem (Iansiti & Levien, 2004).

Within the body of software ecosystems, a slightly different niche player is identified. Bridge players connect different niches or ecosystems with each other. Their value is in the connection they bring and their ability to broker the position. They assist in the information flow of the local innovation system (Kirkels, 2010).

However, the majority of the organizations in the ecosystem can be seen as contributors or complementors to the ecosystem. The niche players can be seen as the leaders in a niche but there can be only one leader in that specialty. The contributors can be second best, or not even close to second best. They develop compatible and complementary products and earn a good living out of it. They might strive to become a niche player or keystone in the future, but haven’t reached it thus far.

The different roles that niche players and contributors can adopt make generalizing findings based on this specie difficult and inconclusive. The application developers, clients, patent holding companies, and network operators can all exist as niche player or simply a contributor. The different species that organizations in a mobile ecosystem can adopt are presented in table 4. In particular platform leaders can be differentiated in two roles whereas the other organizations can adopt different roles.
Table 4: Species in mobile ecosystem and their role in a business ecosystem.

<table>
<thead>
<tr>
<th>Mobile ecosystem</th>
<th>Business ecosystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform leaders</td>
<td>Keystone</td>
</tr>
<tr>
<td></td>
<td>Dominator</td>
</tr>
<tr>
<td>Patent holding companies</td>
<td>Niche player</td>
</tr>
<tr>
<td></td>
<td>Bridge player</td>
</tr>
<tr>
<td></td>
<td>Contributor</td>
</tr>
<tr>
<td>Other developers</td>
<td>Niche player</td>
</tr>
<tr>
<td></td>
<td>Bridge player</td>
</tr>
<tr>
<td></td>
<td>Contributor</td>
</tr>
<tr>
<td>Clients</td>
<td>Niche player</td>
</tr>
<tr>
<td></td>
<td>Bridge player</td>
</tr>
<tr>
<td></td>
<td>Contributor</td>
</tr>
</tbody>
</table>

3.2.3 Variety in species and roles
A greater variety of species and roles can make an ecosystem healthier (Hartigh et al., 2011). It will make the ecosystem more robust and resilient to disturbance and allows for organizations to choose with whom they cooperate, improving the quantity and quality of the relations. Furthermore, a large variety in organizations will foster spawning of creative ideas and can make more and diverse resources available to others (Moore, 1996). However, it must be noted that too many different organizations will slow down the system, increasing the transaction cost of each economic exchange. Furthermore, the slow response will make the ecosystem less capable of responding to adaptations of the environment or external forces (Moore, 2005). A balance in diversity is sought after that is ‘fit’ with its environment (Peltoniemi & Vuori, 2004).

3.2.4 Summary
A large portion of companies in the EU are in the category of SMEs. Their size limits them in their availability to resources but it offers several advantageous characteristics. In comparison with large enterprises, SMEs most often have few decision makers allowing for swift decisions. The organic organization structure stimulates the creation of an informal social system with strong trust bonds. This is helpful in dynamic environments where creating innovative solutions requires close cooperation and sharing of knowledge. Furthermore, the informal social system helps create trust bonds between employees and managers, which can prevent resistance to change in an organization when confronted. Lastly, the trust bonds will help motivate employees and stimulate dedication towards the company.

In a business ecosystem different species can be identified, in particular those of keystones, dominators, and niche players. Next to these species are the contributors or members of the ecosystem. Keystones stimulate the exploration of new business ventures and continuous renewal of the business ecosystem. They offer resources for others in the ecosystem to build complementary products. Their relationships with others in the ecosystem are based on trust through endorsements, certification, and references. As central organization in an ecosystem, keystone organizations can become platform leaders. Dominators try to exploit the ecosystem by gradually taking over all activities by vertical integration, in the role of value dominator. They will try to exploit the existing
business opportunities and prevent radical changes that may influence their value extraction. In the role of value dominator they will try to maximize the value they can deduct from the ecosystem whilst maintaining a heartbeat. A dominator will govern the ecosystem by strict selection of partners and direct account manager contact. Lastly, niche players try to develop unique capabilities or resources that make them ‘special’ in their niche. They leverage complementary resources of other niche players or keystone/dominators for the improvement of the central offering. By constantly innovating they stay ahead of competition and remain a competitive advantage in their niche.

The different species and roles that evolve in the ecosystem help improve the variety of the ecosystem. This in return can improve the overall health and performance of the ecosystem. It allows for organizations to choose with whom they cooperate, improving the quantity and quality of the relations. A balance in diversity is sought after that is ‘fit’ with its environment, preventing that too many different organizations will slow down the system.

### 3.3 Governance

The term governance has different meanings depending on the context within it is used. A meaning of governance that is present in all forms or modes of governance is that of monitoring and controlling relationships and behaviour of actors in a network or organization (Provan & Kenis, 2007) in an attempt to improve coordination between them (Klijn, 2008).

Traditionally, within the body of corporate governance this relates to the role of boards of directors in representing and protecting the interests of shareholders. This is consistent with the principal-versus-agent problem, and is internal to the organization. In public management it relates to organizing public service effectively and efficient. In networks, governance is most often not described as networks imply cooperation between independent actors. However, in a network with a shared goal as in a business ecosystem perspective, it relates to efficiently and effectively organizing collective action to ensure a shared goal is reached (Klijn, 2008). In this chapter the various forms of governance are introduced and in particular governance from a business ecosystem perspective is elaborated upon.

#### 3.3.1 Corporate governance

Corporate governance can be seen as the financial and legal framework for regulating the relationship between a company’s management and its shareholders (Eun, Resnick, & Sabherwal, 2012). It is internal to the organization and finds practice in organizations where the shareholders are not the managers. It becomes most important when there are many different shareholders with small blocks of shares. This is particularly the case with stock market exchanged shares which creates diffused shareholders. The efficiency of risk sharing with a multitude of shareholders has its downside in the possibility of conflict of interest between shareholders and their managers. This is also known as the agency problem. The information asymmetry between managers and shareholders creates a tension that needs to be addressed. Managers have their own agenda in pursuing a career that might not be in the best of interest of the shareholder. This can become evident in excessive wages, or investments in other companies to serve their private interests. It could be prevented if it would be possible to write a contract that would specify exactly what a manager should do. However, in practice this is impossible and managers are given the right to make decisions within a mandate. The mandate is part of the financial and legal framework and with different remedies, or
mechanisms, the agency problem is governed. Next to the mechanisms, that are available to shareholders, national law protects the shareholders.

Various governance mechanisms exist to control the agency problem. Most prominent is a board of directors to represent the interests of shareholders. If the board remains independent of the management team, it can serve as an effective mechanism (Eun et al., 2012). However, in a diffused ownership structure of a public company, managers most often select the board of directors, leading to poor governance results. Other mechanisms are, for example, that of incentive contracts or stock listings in countries that have protection by law that is more to the benefit of the shareholder.

3.3.2 Public management governance
The principles of a properly functioning public administration underlie public management governance. It relates to the fair treatment of citizens and organizations that adheres to the basic principles of the rules of the law (Klijn, 2008). The emphasis is on the operation of government, rather than how it is organized. It is the classical theory of public organization to offer public service. In this notion, the role of the government is to provide public service and to govern these activities in a bureaucracy. Policy making and implementation are vertically integrated within the government. The key governance mechanism is hierarchy, with the focus on line managers and their accountability for spending public money (Osborne, 2006).

During the 90s, this shifted towards New Public Management in which it was not the intention to provide the services but to set goals and to steer other (public) organizations to provide public service (Klijn, 2008). The focus has shifted from implementation towards goal setting. Believing that the activities should be contracted to public or private organizations to provide public services (Hill, 2004). Market mechanisms will assure that the activities are provided efficiently and against competitive pricing. The value base is contained within the belief that the market place is the most appropriate place for the production of public services (Osborne, 2006). Important is controlling the output by clear performance indicators, instead of the process. The governance mechanisms are, for instance, making clear contracts such as public-private partnerships and evaluating in- and output by performance management and auditing.

3.3.3 Relationship governance
Relationship governance concerns with the dyadic inter-organizational relationships that organizations can have. It is different from corporate governance as it is relates to external organizations, and is different from public management governance as it addresses the relationships from a dyadic level of analysis. It refers to analysing a governance structure of the business relationship between a customer and supplier and is based on governance mechanisms of markets, hierarchies, and social mechanism (Kohtamäki, 2010). By applying these mechanisms in a different combination, the supplier or customer can steer the behaviour of the other organization (Adler, 2001).

Recent research is on the judgement of organizations in the relationship to analyse the stability of the relationship and predict behaviour from it (Ritter, 2007). Previous research has focused on relationship learning or on governance structures, but not on the effect of governance structures on relationship learning (Adler, 2001; Nooteboom, 2004). Scholars therefor try to assess the impact of governance structures on organizational learning in partnerships between organizations to drive innovation (Kohtamäki, 2010).
Relationship governance is in particular useful for analysing the relationships and identifying the mechanisms that are used in a relationship. It can be used to develop strategies to predict behaviour and can help organizations optimize the value they extract from relationships. However, as it only analyses the dyadic relationships, it lacks notion of the environment. In particular the environment of an organization determines what partners to select and how to proceed in the relationship as new information can change the decision heuristics of the people involved.

3.3.4 Relationship governance in a mobile ecosystem

In a mobile ecosystem all actors involved collaborate, compete, and co-evolve towards a shared goal. Mobile ecosystems are not legal entities and there is no imperative for organizations to govern the ecosystem by themselves. However, to reach that goal, collective action must be organized efficiently and effectively (Provan & Kenis, 2007). As the actors are most often independent firms they typically have limited formal accountability and conforming to rules and procedures is voluntary (Provan & Kenis, 2007). There is a lack of hierarchical power, making managing the business ecosystem impossible. Instead, the actors try to govern the business ecosystem (den Hartigh & Tol, 2008).

In a literature survey of Jones et al., several definitions of governance in a (business) network are summarized. From this overview a new definition has been developed that offers a bottom-up approach in governing relationships (Jones et al., 1997). The definition they have developed is followed in this research. Relationship governance in a mobile ecosystem “involves a select, persistent, and structured set of autonomous firms (as well as non-profit agencies) engaged in creating products or services based on implicit and open-ended contracts to adapt to environmental contingencies and to coordinate and safeguard exchanges” (Jones et al., 1997, p.914). What are governed are the exchange relationships between the actors but in the context of the interdependencies between the actors in the ecosystem. Each individual exchange relationship can be influenced by the participating actors but also by other members of the ecosystem. It can be seen as a micro-economic approach to economic exchanges. What is important are the actors and the conditions under which the exchange takes place. These can be influenced by governance mechanisms. In that sense, it incorporates the method of analysing relationships from relationship governance, but makes notion of, and involves, the environment surrounding the relationship.

It is distinct from a top-down approach as described by Provan and Kenis (2007). They argue that in network governance there are actors in the network or representatives of the network that control and oversee the network. These actors ensure that visions are aligned and the actors have the right incentives to do business with each other. It can be seen as a macro-economic approach to economic exchanges on a strategic level of the business ecosystem.

3.3.5 Summary

There are different interpretations of governance, depending on the context and perspective from which it is adopted. In general governance can be seen as an activity of monitoring and controlling relationships and behaviour of actors in a network or organization (Provan & Kenis, 2007) in an attempt to improve coordination between them (Klijn, 2008).

Within a corporation this is done with the financial and legal framework for regulating the relationship between a company’s management and its shareholders. This is necessary as the information asymmetry between managers and shareholders creates a tension that needs to be
addressed. Public management governance relates to a properly functioning public administration and the fair treatment of citizens and organizations that adheres to the basic principles of the rules of the law. In this notion, the role of the government is to provide public service and to govern these activities in a bureaucracy. Policy making and implementation are vertically integrated within the government. When analysing on a micro-economical level relationship governance can be used to analyse the mechanisms behind the relationship and predict behaviour from it. However, it lacks a notion of the environment as the focus is dyadic and thereby is unable to grasp the dynamics in a larger business network.

In this research the perspective of an application developer (as SME) is adopted that tries to influence the relationship it has with other organizations in the ecosystem. What is governed are the exchange relationships between the organizations with respect to the interdependencies between them. It is a bottom-up approach to exerting governance mechanisms in a relationship and in that sense is distinct from other modes of governance. Relationship governance in the context of a mobile ecosystem can be seen as an attempt to influence and shape the relationships in a business network by adapting, coordinating, and safeguarding the economic exchanges (Jones et al., 1997).

3.4 Transaction cost economics
Transaction cost economics (TCE) can be seen as a first attempt in analysing economic exchanges, or transactions, instead of the goods or services, deviant from the neoclassical view of the firm. The transaction cost economics principle has been introduced by Coase in 1937 but has become widely known by Williamson in 1979 by defining the cost of transactions in making or buying a product. If the costs of producing in-house are higher than the market, a buy decision is made and if producing in-house is cheaper than the market a make decision is made. The market or hierarchy mechanisms of production are the two primary dimensions of TCE. Market governance mechanisms are used when the activities are organized external to the firm and hierarchy governance mechanisms are used when the activities are organized internal to the firm (Williamson, 1979). Legal contracts and prices are the coordinating mechanisms in market governance, whereas employment contracts and authority are the coordinati ng mechanisms in hierarchy governance (de Reuver, 2009). According to Williamson (1991), the organizations within TCE are assumed to be rationally bounded, risk neutral, and at least some are opportunistic to allow for competition.

Research on governance has most often taken a dyadic perspective on organizations. As discussed this offers a limited view on the economic reality. Current research is taking a more broad perspective in networks (Provan, Fish, & Sydow, 2007), whole networks (Ahuja et al., 2011), value networks (de Reuver, 2009), and in business ecosystems (Iansiti & Levien, 2004; Kapoor & Lee, 2013; Wareham et al., 2012).

Since its conception TCE has received criticism. Most criticised is the omission of the social embeddedness of economic transactions (Granovetter, 1985), which can be both structural (Granovetter, 1985), as well as relational (Gulati, Nohria, & Zaheer, 2000). But also the dimensions have been subject of criticism, being mutually exclusive in its original form. It has led to additions of a ‘network’ dimension as hybrid form in between markets and hierarchies (Williamson, 1994), but also as distinct form that “contrasts and competes with markets and hierarchies” (Jones et al., 1997,p.914). Recent research is on the simultaneous use of the governance mechanisms from the three dimensions, being non-mutually exclusive. The empirical evidence has shown that in reality the
mechanisms that are used in the dimensions coexist and are used next to each other (Bradach & Eccles, 1989; de Reuver, 2009; Lowndes & Skelcher, 1998; von Tunzelmann, 2003). As for instance, authority mechanisms frequently exist in written contracts, or are implicitly present within an industry or network. Therefore, throughout this research the simultaneous use of the different governance mechanisms is followed. In the following paragraphs the different dimensions and governance mechanisms are discussed. The exchange conditions under which a certain dimension is more, or less, useful will be discussed in a later paragraph.

3.4.1 Hierarchy and authority
The classic dimensions from TCE are hierarchy and markets being mutually exclusive in its original form. Market failure justifies the need to organize activities internal to the firm based on the hierarchical mechanism of authority. The ability to exercise control can be seen as the primary reason for hierarchical mechanisms to be more efficient than market mechanisms. This is reached with an employment relationship or contractual arrangement that provides decision-making authority in certain areas (Geyskens et al., 2006). Between organizations it is not an employment relationship but a difference in power that gives an organization the decision-making authority (de Reuver, 2009). As in politics, the power distance between organizations or individuals can be influenced by lobbying. It can be costly as it lowers the quality of the decision and can divert attention from more productive activities. Authority can be seen as the extent to which an organization can influence the decision making process of another organization (de Reuver, 2009).

3.4.2 Markets and contracts
In markets it are contracts and prices that are used to organize collective action. As stated by Powell (1990), markets “are a spontaneous coordination mechanism that imparts rationality and consistency to the self-interested actions of individuals and firms” p.302. The market is open to new participants and is free of future commitments. It are the prices that determine the success of an exchange as it is most often the strategy to drive for the best bargain (Powell, 1990).

The governance mechanisms of the market dimension are contracts and prices. They offer choice, flexibility, and opportunity, as a device for fast and simple communication (Powell, 1990). The assumption is that the market is more efficient than vertical integration due to competition (Williamson, 1979). Vertically integrated organizations can become bureaucratic and slow, increasing the costs of a transaction. However, certain exchange conditions can create market failure, making hierarchical or network governance mechanisms more efficient as will be described in section 3.5.

Contractual based governance helps reduce the hazards in a partnership as they provide an efficient safeguard against opportunism (Williamson, 1979). This is achieved by placing limits to the actions of partners and enhancing monitoring (Hoetker & Mellewigt, 2009). Furthermore, they facilitate the control of information shared between partners and set out the rules for the resolution of any dispute that might occur (Poppo & Zenger, 2002; Teng & Das, 1998). They have, however, the difficulty that everything should be known beforehand a project, to allow the drafting of a contract. In particular in high-technology environments with unknown outcomes of a development project, this is difficult to write down (Bradach & Eccles, 1989). The contract is used to draft up specification, conditions, and the price for which the exchange will take place. It involves the division of benefits and guarantees that can be given upon delivery. It is dependent on local law and the assumption of fair competition.
3.4.3 Networks and trust

The boundaries of a firm in an economic context have blurred as they engage in collaboration that does not resemble market like transaction nor a hierarchical one. In a seminal work by Powell (1990) he argues that it are relational contracts that are increasingly becoming important in a transaction instead of formal contracts or bureaucratic structures. The relational contracts are characterized by informal social systems that help firms meet resources and functional needs from their network instead of from vertical integration or by the market (Powell, 1990). Conflicts are resolved by reputational concerns instead of law enforcements and the means of communication are relational instead of by pricing. In markets a firm will try to bargain the best deal while in a network it will try to create indebtedness and reliance in the long run. The bureaucratic structure of hierarchies foster efficient decision making and clear departmental boundaries. The strength of hierarchies are in reliability and its accountability for efficiently exploiting activities. But when uncertainty disturbs the environment, the liabilities of hierarchies are exposed (Powell, 1990).

The governance mechanisms of the network dimension, also referred to as relational mechanisms (Geyskens et al., 2006), are based on the trust in a relationship that each party will live up to the requirements. Reputation, goodwill, and referrals are important drivers that help build up trust between the organizations. The basic assumption of a network of relationships is that one party is dependent on resources of another and that pooling of these resources can lead to gains. They put in effort to build up a relationship and over time it becomes economically sensible to exercise voice rather than exit (Powell, 1990). Voice further stimulates the creation of friendship, reputation, and interdependence, integral parts of the relationship. Information from friends, or someone you have dealt with in the past and has proven reliable, is often more valuable than from strangers. The information that flows through networks is therefore more ‘thicker’ than information from markets, and ‘freer’ than that from hierarchies. This makes networks in particular useful for the exchange of commodities whose value cannot easily be measured (Powell, 1990), such as with innovations or uncertainty in the technology. Powell concludes that network mechanisms are in particular useful for collective action. In which cooperation can be sustained over the long run as an effective arrangement. The relationships in this cooperation create incentives for learning and the dissemination of information. This will improve the speed of translating ideas into action. With variable resources and high environmental uncertainty networks are most useful as they offer feasible means of using and improving tacit knowledge and technological innovation (Powell, 1990).

The dominant governance mechanism in the dimension of networks is trust. It is “the willingness to accept vulnerability based on positive expectations about another’s intentions or behaviours” (McEvily, Perrone, & Zaheer, 2003, p.92). To understand network-level interactions, it is the distribution of trust that is critical and if it is reciprocated among the network’s members. It can be widely distributed, i.e. a high density of trust relations, and it can be narrowly distributed, i.e. low density of trust relations. A distinction is made between interpersonal and institutional trust. Interpersonal trust relates to the trust that is obtained directly from individuals or groups. Whereas institutional trust is experienced or observed indirectly by work or observing quality of institutions.

Interpersonal trust can be seen as a process-based trust that relies on a personal relationship between two actors and is therefore highly particularistic. Strong ties are ties that are emotionally intense and supportive, facilitate mobilization of resources and are marked by high levels of interpersonal trust. The level of trust decreases as we move from strong interpersonal ties with our
family to anonymous ties with strangers, people on the street, etc. Strong ties tend to occur in network structures that are characterized by a high level of closure or network density, which allow for the exercise of reinforcement of positive expectations by means of close monitoring and social control (Rus & Iglic, 2005).

Mutual trust between organizations, or inter-institutional trust, can be seen as the reliability that the other organization will fulfill its obligations. Trust creates the expectation that the organizations will show predictable behaviour according to the agreement. The expectations thereby reduce transaction costs, as less monitoring and renegotiating the exchange is needed when environmental changes occur (Jones et al., 1997). In particular in highly complex tasks facing strong time constraints (Jones et al., 1997). Furthermore, it is believed that trust affects the depth and richness of exchange relations, particularly with respect to the exchange of information. Trusting behaviour is cited as a critical factor for improving innovation through collaboration (Hausler, 1994).

The exchange conditions as described under markets and hierarchies remain in the network form. Rather, research has focused on a more detailed description of the conditions under which the network form prevails. Powell (1990) mentions factors that make the network mechanism prevail over the other mechanism. Jones et al. (1997) use the previous mentioned exchange conditions and add complexity of tasks.

3.4.4 Summary
Transaction cost economics can be seen as a first attempt in analysing economic exchanges, or transactions, instead of the goods or services. It is defined as the cost of transactions in making or buying a product. If the costs of producing in-house are higher than the market, a buy decision is made and if producing in-house is cheaper than the market a make decision is made. According to Williamson (1991), the organizations within TCE are assumed to be rationally bounded, risk neutral, and at least some are opportunistic to allow for competition. The original dimensions of markets and hierarchies by Powell (1975), have received criticism which has led to the addition of a distinct network dimension. Different governing mechanisms are used in the dimensions.

Price mechanisms and contracts are the governing principles of markets. Due to competition an organization can achieve the best deal in the market. A contract is used to specify the conditions of the work and offer an exit in case of any dispute. But as the frequency of the transactions increase, the market becomes less efficient and hierarchical mechanisms prevail. By vertical integration the costs of transacting can be reduced as authority replaces complex contracts and renegotiation. Between organizations it is the power distance that gives the authority to decide. Large organizations on which the partner is dependent on, give control over the relationship. A similar mechanism is present with specialists and their control over assets. But when uncertainty disturbs the environment, the liabilities of hierarchies are exposed making relational governance mechanisms more efficient. It is based upon the trust that each party will fulfill its obligations. Trust creates the expectation that the organizations will show predictable behaviour according to the agreement. The expectations thereby reduce transaction costs, as less monitoring and renegotiating the exchange is needed when environmental changes occur.

Recent research is on the simultaneous use of the governance mechanisms from the three dimensions, being non-mutually exclusive. The empirical evidence has shown that in reality the
mechanisms coexist and are used next to each other. Following the line of reasoning, throughout this research the simultaneous use of different governance mechanisms is assumed.

3.5 Exchange conditions
Exchange conditions of transactions raise transaction costs and can create market failure (Geyskens et al., 2006). Leading to the decision to produce internal to the firm or integrate vertically. These conditions are; ‘asset specificity’, ‘uncertainty’, and ‘transaction frequency’ (Williamson, 1979). They are the conditions surrounding the transaction in the dimensions of markets and hierarchies and are the observable measures as identified by Williamson (1979). Following the reasoning of Jones et al. (1997) the exchange conditions of the network dimension are similar but have the additions of ‘task complexity’ and are more specific to the situation in which the network form will emerge and thrive. In the following paragraphs, the different common conditions are described.

3.5.1 Asset specificity
Exchanges that have been customized, or entail specific assets, create dependency between partners (Jones et al., 1997). A high asset specificity of a transaction involves assets that are tailor made to the transaction. There is a safeguarding problem as market competition will exploit the assets opportunistically, increasing the transaction costs. The authority relationships and hierarchical control procedures through vertical integration are assumed to embody greater safeguarding capabilities and are seen as the solution to the problem (Geyskens et al., 2006).

Firms create these safeguarding capabilities to protect themselves from opportunistic behaviour. With high levels of asset specificity, opportunistic behaviour is likely to occur (Williamson, 1979). Furthermore, large investments that are sunk costs for an organization, will give an incentive to other organizations to behave opportunistically. As a result, instead of market mechanisms, hierarchy mechanisms are chosen, by acquiring the other organizations (van de Vrande, Lemmens, & Vanhaverbeke, 2006).

Organizations that offer services based on intellectual capital or craft-based skills usually have a lot of know-how which has been developed over the years. It typically involves tacit knowledge that is difficult to codify and can be seen as intangible and mobile assets. They are present in the minds of experts and specialists and are less subject to be dictated by a higher authority and might walk away to another company if the price they receive is better there. Rather, the distinctive competencies are more likely to be kept in the transaction when a network mechanism is used (Powell, 1990). The trust bonds will stimulate knowledge sharing and a certain commitment of the specialist to the other party. Hence, intangible assets, with high human asset specificity (Jones et al., 1997) are more likely to be governed with network mechanisms, whereas tangible assets such as equipment, patents, and regular services are more likely to be governed by market mechanisms (Powell, 1990). This distinction is elaborated on by Hoetker & Mellewigt (2009), who states that property-based assets that are easily codified will be better suited to formal controls. The inability to specify processes and outcomes in advance make relational governance mechanisms with knowledge-based assets better suited (Hoetker & Mellewigt, 2009).

3.5.2 Uncertainty
A second condition is that of uncertainty which arises when project properties are too unpredictable to be specified beforehand in a contract, or the performance cannot be verified afterwards (Geyskens et al., 2006). It can be broken up into environmental uncertainty and behavioural
uncertainty. With relational governance mechanisms, less specified contracts are needed to achieve the outcome of a transaction. It is based under the assumption that contingencies will be addressed in good faith and shirking will not occur (Kogut, 2000). The costs occurred due to the difficulties of ex ante describing contingencies in a contract are therefore reduced when relational mechanisms are used (Hoetker & Mellewigt, 2009).

The environmental changes that require adaptations to an agreement raise transaction costs and can be solved by hierarchical mechanisms of a contract or authority. However, uncertainty is too broad a concept to define it in one direction. High environmental uncertainty can also be resolved by remaining flexible (Geyskens et al., 2006), a property in contrast with hierarchical mechanisms. Therefore, in a study by Walker and Weber (1984), environmental uncertainty has been broken up into volume uncertainty and technological uncertainty, extending the framework by Williamson (1979). Volume uncertainty involves not knowing what resources are needed in advance of the transaction. As a result, suppliers can be left over with excessive stock or unexpected production costs. Buyers can experience a shortage in the market or again excess inventory. A solution would be to produce the products internally as the firm should be able to organize this more efficiently. The hierarchy mechanism would prevail the market mechanism. Technological uncertainty occurs when the technological requirement in a relationship are unknown beforehand (Walker & Weber, 1984). Following from uncertainty in standards or specification of components, but also from general technological development. Unlike volume uncertainty, flexibility in a relationship will improve the ability of the firm to cope with these uncertainties. A market mechanism will be more suitable than a hierarchical mechanism. It will prevent lock in to a supplier and allow a firm to switch to different suppliers more easily if their capabilities are more appropriate (Balakrishnan & Wernerfelt, 1986; Heide & John, 1990).

The behavioural uncertainty surrounding a transaction involves the difficulties in evaluating if the performance measured ex post meets the expectations. The classical argument from TCE would be to integrate the activities vertically in such a situation, a make decision is then made. This will allow the firm to remain in control over these activities and acquire more complete information for the ex post evaluation (Geyskens et al., 2006).

A key benefit of networks for the entrepreneur is the access they provide to information and advice. Often only a few contacts provide the multiple resources. Apart from resources, the relationships in a network can also have a reputational or signalling effect (Hoang & Antoncic, 2001). When organizations are attempting to enter into a relationship with each other, they seek legitimacy by trying to reduce the perceived risk. By associating with, or seeking certification from, well-regarded individuals and organizations, they can help reduce this uncertainty (Hoang & Antoncic, 2001). Trust between partners is perceived to enhance the quality of resources (Hoang & Antoncic, 2001). The social mechanisms of power and influence, and the threat of ostracism and loss of reputation can create cost advantages in comparison to market or hierarchy mechanisms (Hoang & Antoncic, 2001). Trust is used as cost efficient mechanism in this sense.

The social mechanisms of reputation and collective sanctions safeguard specific exchanges in the network by dispersing information about behaviour and increasing the costs of inappropriate behaviour. Collective sanctions reduce behavioural uncertainty by increasing the cost of opportunism and by decreasing the costs of monitoring to any individual party. Similarly, reputations which have
economic consequences for participants in networks reduce behavioural uncertainty by providing information about the reliability and goodwill of others (Geyskens et al., 2006).

In a relationship or transaction that strives for innovations, coordination is essential as it reduces uncertainty and opportunistic behaviour (Dhanaraj & Parkhe, 2006). Access to information, flexibility, and responsiveness to changing environments makes a network form of governance able to offer this coordination (Powell, 1990). Passing information up or down a hierarchy or purchasing it in the market gives control over existing information but not on new combinations. In a network form of governance the trust bonds between the organizations allow for new meanings to be generated, debated, and evaluated (Powell, 1990).

3.5.3 Frequency of transaction
A third condition is that of the frequency of transaction. It involves the extent by which the transaction reoccurs between the organizations. If there is a high frequency of transactions a hierarchical mechanisms will prevail the market mechanisms. The organizations have an incentive to make contractual agreements that are more efficient than recurring market mechanisms (Williamson, 1979). Furthermore, an increased frequency of transaction will reduce the information asymmetry between the organizations making hierarchical mechanisms more suitable (van de Vrande et al., 2006). It can also be argued that this will improve the trust levels between the organizations which in turn will improve the network mechanisms (van de Vrande et al., 2006). Geyskens et al. (2006) point out in their meta-analysis that this characteristic has received little attention in scientific literature in comparison with asset specificity and uncertainty. And although in literature asset specificity is seen as superior to uncertainty, the results of their meta-analysis showed no distinction of importance between the two.

Repeated interaction among individuals has been demonstrated to have powerful consequences. When it is highly likely that individuals will meet again in the future, they are not only more likely to cooperate with others but also willing to punish those who do not cooperate (Axelrod & Dion, 1988). With frequent transactions, quality becomes more important than quantity. The reputation that an organization is able to meet up to the requirements and maintain or improve the level of quality is an important driver in the relationship. As a result there is little need for hierarchical mechanisms as the desire to continue the collaboration is stronger than the opportunism to destroy it (Powell, 1990).

3.5.4 Task complexity
The complexity of a task can be described as the different specialized inputs needed to complete a product or service (Jones et al., 1997). It requires more coordination and creates behavioural interdependence between the organizations. It can result in factors such as increased scope of activities, number of functions needed, or increase of number of markets (Killing, 1988). By use of mutual adjustment between participants, through information flows and meetings, these difficulties can be overcome. The network governance form stimulates this behaviour and is likely to be preferred over other governance methods.

3.5.5 Summary
Depending on the exchange conditions surrounding a transaction, organizations use different governance mechanisms. The conditions can be summarized in four categories; asset specificity, uncertainty, frequency of transaction, and complexity of tasks.
With respect to hierarchies and markets, asset specificity, volume uncertainty, behavioural uncertainty, and frequency of transaction promote hierarchical mechanisms. Whereas technological uncertainty prefers market mechanisms. However, as asset specificity, technological uncertainty, frequency of transaction, and complexity of tasks increases, network governance is preferred over hierarchy governance. As volume uncertainty increases, market governance is preferred over network governance.

In practice, the governance mechanisms are used simultaneously as for instance a contract will have both authority as well as price mechanisms, and the partners in the contract will trust each other in fulfilling their obligations.

### 3.6 Conceptual framework

When looking at a business network from the perspective of a mobile ecosystem distinct roles and species become visible. In general, the network revolves around a platform leader that has developed a platform. The platform leader offers the platform to users and organizations that wish to participate (Gawer & Cusumano, 2008). The boundary of the ecosystem are the organizations that are involved in the platform by offering compatible and complementary products (den Hartigh & Tol, 2008). An organization is not retained to one platform and can be part of multiple ecosystems.

According to their size, the organizations can be categorized into small, medium, and large enterprises. The majority of organizations are small- to medium sized enterprises (European Commission, 2005), that take on the role of contributor or niche player in the ecosystem. The platform leaders are most often large enterprises that differentiate their role into keystones or dominators (Gawer & Cusumano, 2008; Iansiti & Levien, 2002). The mobile application developers often have relationships with platform leaders, clients, other developers and in some occasions patent holding companies.

In a business ecosystem all organizations involved collaborate, compete, and co-evolve towards a shared goal (Iansiti & Levien, 2004; Provan & Kenis, 2007). However, to reach that goal, collective action must be organized efficiently and effectively (Provan & Kenis, 2007). Business ecosystems are not legal entities and there is no imperative for organizations, other than platform leaders, to govern the ecosystem by themselves on a strategic, ecosystem wide level. Furthermore, SMEs have limited resources (Welsh & White, 1981) and a dependency on the platform leader (Gawer & Cusumano, 2008) which limits them in performing governance. It is the platform leader who in general tries to orchestrate collective action (Dhanaraj & Parkhe, 2006). However, the high degree of interdependence in the ecosystem requires smaller organizations to work together with other organizations if they are to deliver new innovations (Ghobadian & Gallear, 1997). Hence, the SMEs engage in a relationship with other organizations to develop the compatible and complementary products. The relationship can be seen as economic transactions, or exchanges, that can be influenced through governance mechanisms (Williamson, 1979), as a bottom-up approach in governing exchanges.

Transaction cost economics can be used to analyse the relationships or economic exchanges, instead of the goods and services (Williamson, 1979). With the dimensions of markets, hierarchies, and networks as distinct and non-mutually exclusive (de Reuver, 2009; von Tunzelmann, 2003). Depending on the exchange conditions surrounding the transaction, certain governance mechanisms will see more practice as they are more effective or efficient than others (Jones et al., 1997; Powell,
1990; Williamson, 1979). The exchange conditions are dependent on the role or specie, and the size of the organization, with whom the transaction is commenced (Pittino & Mazzurana, 2013).

This research tries to identify the governance mechanisms that are, or can be, exerted by mobile application developers in a mobile ecosystem as sub-group of business ecosystems. With the goal of the developer to influence the relationship with the other organization. The role, or specie, of the other organization, will influence the exchange conditions. As a result, from the perspective of the developer, certain governance mechanisms will be more useful than others. From a case study the relationships between the concepts will be explored. This will allow for the development of hypotheses that can be tested in future research.

A conceptual framework of the relationships under research is given in Figure 3. It shows how the relationship a developer can have with the different species in the mobile ecosystem can be based on different governance mechanisms. Each relationship can be analysed with transaction cost economics to identify what governance mechanisms are used which is influenced by the exchange conditions surrounding the transaction.

![Figure 3: Conceptual framework of the relationships under research.](image-url)
4 Method

The research has been performed as a single-case study from the perspective of an application developer that is active in the mobile ecosystem of either Apple or Google (or both). The unit of analysis are the governance mechanisms that are adopted by the application developer depending on the specie in the relationship. This has resulted in four embedded cases as the developer enters into a business relationship with a platform leader, clients, other developers, or patent holding companies. The details of the method are elaborated upon in this chapter. This research aims to derive testable hypotheses for each embedded case and contribute to the general understanding of relationships between application developers and other organizations in a mobile ecosystem. By exploring the mechanisms in a relationship between these organizations, this is achieved.

Different qualitative research methods exist that are more or less suitable depending on the situation. The newness of the mobile ecosystem and the application of the theory in this domain, make a case study approach a preferred method. It is seen as a powerful research instrument to explore phenomena in their real-life setting. Research in this domain from the chosen perspective is scarce and needs to be investigated in its preliminary stage (Eisenhardt, 1989). And as the research question addresses ‘why’ and ‘how’ phenomena occur, an exploratory research method is most suitable (Yin, 2011). Furthermore, the case study research can provide in flexibility to steer the research as new knowledge is identified during research (Yin, 2011). It offers the researcher the possibility to explore concepts and the varying perceptions that practitioners can hold.

Nevertheless, case study research can suffer from structural problems in the way the research is performed. If not addressed properly it can reduce the validity and reliability of the research, limiting the generalizability of the findings (Yin, 2011). Also, given the time constrains in this research, no longitudinal data can be collected, although the business ecosystems is constantly evolving by nature.

In the following section the research design is discussed. The data collection method and analysis are described in the next sections. Validity considerations are discussed to assure the generalizability of the findings. A summary concludes the chapter.

4.1 Research design

The research design consists of a single-case-study approach with four embedded cases. The organizations with who the mobile application developers has business relationships in the platform of both Google and/or Apple are studied. A multi-case study approach in different industries would yield more generalizable results but is given the time constraints not possible in this research.

Google and Apple are the platform leaders in the respective mobile ecosystems, and are the dominant mobile platforms in the current market. Their popularity reflects in the amount of developers that participate in the ecosystem.

The relationship the developers have with other species in the business ecosystem is the unit of observation. It allows to identify to what extent SMEs can influence their relationship with other organizations and by what governance mechanisms this is achieved. The unit of analysis are the governance mechanisms that are used between the developer and other species in the business ecosystem in the process of development of mobile applications. The boundary of the case study is surrounding the platform leaders and organizations that deliver contributions or complements to the
core product of the platform leaders. A developer can be active in both the ecosystem of Apple as well as Google.

The process of the research is divided in three consecutive and distinct phases. With use of information from literature the different concepts that are topic of this research are explored in the first phase. The relevant concepts are described in the theoretical framework. The connections between the concepts are discussed in the conceptual framework. It determines what is in scope of this research and is the starting ground for the exploration of the domain. Based on the conceptual framework and the findings from the domain, the questions for the interviews and the interview protocol are developed.

In the second phase the preparation for collecting data is performed as well as the actual collecting. With use of the concepts from the conceptual framework, the questions for the semi-structured interviews have been developed. The case study questions give the researcher general guidelines on how to collect the data. The data is collected by performing semi-structured interviews and observations. The interviews have been transcribed and have led to an in-depth case description of developers in a relationship with other types of species in the mobile ecosystem. The analysis of the in-depth case description finalizes this phase.

From the analysis of the in-depth case description, in the third phase the hypotheses are formed with respect to the conceptual framework developed in the literature analysis. They are subject to future research. The answer to the research question, and sub-questions, is given in this phase from which the conclusions for the research are drawn. Recommendations, limitations, and a reflection finalize the thesis.

4.2 Data collection

Information with respect to the business relationship of developers is often not written, let alone publicly available. Managers of SMEs find documenting their activities in general a waste of time and as a result, there is little written information. Therefore the main information sources for the case study are interviews and observations. They are enriched with information from secondary sources such as web pages and forums. Those sources can be valuable in giving up-to-date background information of the mobile ecosystems.

4.2.1 Interview protocol

Collecting data for the research involved using the typical qualitative research methods of interviews and observations of which a summary can be found in appendix C. The conversations were not recorded and held at location of the interviewees preference. Thereby it was possible for them to express their selves in a ‘real’ setting and not feel uncomfortable. This allowed true observations to be made in their actual environment, within their social system. Immediately afterwards the findings were summarized and transcribed based on the questions that were asked. This allowed the interviews to be structured and comparable with each other. General questions were developed with themes of the concepts from the conceptual framework. Follow-up questions were asked to further explore concepts. After each interview, the questions have been revised to correct possible misunderstandings and to achieve more accurate answers. The interview protocol with questions can be found in appendix A. To validate the findings and reduce errors, the summaries of the interviews were sent to the interviewee for approval.
Questions for the interview were developed with the conceptual framework as a guide. From the literature analysis the relevant concepts that needed to be explored have emerged as themes. The questions were chosen in a sequence that allowed the interviewee to get accustomed with the material and allow a deeper understanding of the concepts while the interview progressed. It started with a broad exploration to identify to what extent the interviewee is aware of the concepts of business ecosystems and governance. It gave an indication of the ability of the interviewee to respond to questions that had a high conceptual level. To prevent bias occurring when explaining concepts, questions have been chosen to be as fool-proof as possible to make it possible for a broad audience to participate. The questions described the concepts in language that is common to the interviewees. As a result, interviewees were able to respond and provided in valuable information for the case study.

The second theme explored the business network of the developer by exploring what business relationship he/she has. Furthermore, it gave insight into the position that the organization perceived to have in the mobile ecosystem and the other organizations that are active in the mobile ecosystem.

The third theme explored the exchange conditions surrounding the relations, and gave insight into what governance mechanisms would be preferred based on theory. The conceptual concepts of this theme have been translated into everyday language to allow the interviewees to respond. In a similar fashion has the fourth theme been developed. It gave insight into the governance mechanisms that the interviewees actually use and explored possible differences with respect to theory and practice. The fifth theme finalized the interview and left room open for input from the interviewee.

4.2.2 Sampling
The interviewees were found by contacting a key informant in the community of application developers. This allowed for a snowballing technique to find possible interviewees and events that could be observed. The interviewees should be either a developer of mobile applications or involved in business with respect to mobile application development. The process of finding new interviewees was stopped when no new insights were given. Typical functions or roles of the interviewees were; (self-employed or employed) developers, CEO (of smaller companies), and project managers.

To improve the validity of the findings a different perspective was sought for after the interviews with the developers gave no new insights. By interviewing clients of developers, and developers of websites and software, instead of mobile applications. This further validated the findings and offered new insights in the mechanisms that are used in the relations. An overview of all the interviewees can be found in appendix B.

The interviewees were offered a summary of the practical findings of this research. A slide deck is made that entailed practical recommendations for the practitioners.

4.3 Data analysis and validity
The data analysis is performed in light of a qualitative content analysis. Following the case study method as proposed by Yin (2011), a method of pattern matching is used. With this method the gathered data from the two cases is organized according to issues that have emerged, and the conceptual framework.
4.3.1 Data analysis
The patterns have been identified in the conceptual framework and compared with the results of the case study. Categorization has been reached by clustering the units of observation. With all the data processed and summarized in the analysis, the clusters show the overall perceptions of the developers for each category. This method is focused on finding answers to the research question (and sub-questions), and avoids drifting off to areas beyond the scope of this research (Yin, 2011).

The case study is developed in line with the conceptual framework. Four groups were identified that were explored with the information sources relevant to the case. These are; platform leaders, clients (of developers), other developers, and patent holding companies. Substance was given to the case study by adding information from the interviews, and observations. If additional information was needed to clarify concepts, secondary sources were added to give context to the subject matter. In each section of the case study, important topics that emerged during the interviews have become paragraphs.

Once the case study had been developed, pattern matching was performed to further identify the concepts from the conceptual framework. By comparing the concepts from the conceptual framework with the case study, conditions, mechanisms, species and other relevant concepts were identified. The concepts that were identified are used to develop hypotheses for each embedded case.

4.3.2 Validity
The quality of the research can be improved by assuring the validity and reliability of the study. As mentioned, a case study research is susceptible for a lack of structure and faces difficulties in replication. This can limit the generalizability of the findings to the case study if not addressed properly. Four tests have been proposed with regard to validity and reliability that can improve the overall quality of the research (Yin, 2011). Qualitative research can be subject to bias from the researcher (Eisenhardt, 1989) and applying these tests can help prevent this from happening.

The internal validity is improved by having used a case study protocol to guide the process of research. The case study protocol incorporated an interview protocol, literature database, and securing interesting links to people and information for acquiring data for the research. The data analysis method has improved the possibility to replicate the data analysis and validate the research. The external validity is improved with use of a theoretical framework and description of the domain. To test the validity of the findings in this context, interviews with developers from similar but different environments have been performed. The transcribed interviews and validation from interviewees have improved the construct validity. Observations have confirmed findings from the interviews and the validation by interviewees after the interview have supported the construct validity. Also, these activities have created a chain of evidence. Lastly, by implementing the above mentioned test, the overall reliability of the research is improved.

4.4 Summary
The method that is adopted in this study is a single-case-study research with four embedded cases. The newness of the mobile ecosystem and the application of the theory in this domain, make a case study approach a logical method. It is seen as a powerful research instrument to explore phenomena in their real-life setting. The unit of analysis are the governance mechanisms that are used between
the developer and other species in the business ecosystem in the process of development of mobile applications.

The process of the research is divided in three consecutive and distinct phases. In the first phase the literature analysis is performed resulting in a conceptual framework. Furthermore the domain is described in which the research is performed. It is essential input for the second phase in which, based on theory and domain, the interview questions have been developed. With these questions the data is collected and observations are made. The third and final phase consisted of the analysis of the data and finding answers to the research question(s).

By following a case study method as described by Yin (2011), the validity and reliability of the research can be assured. This is to the benefit of the overall quality of the research and the possibility to generalize the findings.
5 Case study

The mobile ecosystems from the perspective of application developers have been explored in a single case-study via semi-structured interviews and the participation at offline community events. Four embedded cases have resulted as the developer enters into business relationships with different participants in the mobile ecosystem. The results of these interviews and events are summarized in this chapter.

This chapter describes the findings from the interviews that have been conducted and the observations of the events. These findings are enriched with secondary sources such as web pages and information from forums. The conceptual framework is used to present the findings of the case study. From the perspective of the application developer, the relationships are clustered to the participants in the mobile ecosystem. According to the identified species of participants, these are: the platform leader, the client, application developers, and patent holding companies.

The interviews have been conducted during private interviews as well as developer events, in particular Appsterdam and Cocoaheads. The observations were made during these events. The research has been conducted over the period of May 29th 2013 until July 2nd 2013. A specification of the interviewees and observations is found in appendix B. The summaries of the conversations are found in appendix C. The findings have been clustered according to the organization with who the developer has a business relationship, following the conceptual framework.

5.1 Application developers and platform leaders

The platform leaders provide the platform for the mobile ecosystem. They develop the operating system on which the smartphone is running and develop complementary goods for the participants of the ecosystem. Application developers use the Software Development Kit (SDK) for the development of mobile applications. And contribute to the platform by developing complementary applications. Choosing a platform to develop for requires the use of different business models and handling entry barriers. In this section it is elaborated upon.

5.1.1 Developing applications

The applications for the platforms are developed with the SDK, it is accompanied with development guidelines. In the ecosystem of Apple this is obligatory, not following these rules will lead to an application that will not be admitted to the App store (Kokabha, 2012). In the ecosystem of Google it is not obligatory, although not complying with the guidelines will in general result in an application that is less consistent with the Android platform. As Koen Mostert (KM) mentions, following these guidelines will improve the consistency of applications on the Android platform and improve the overall Android experience. Google also has admission criteria but is much less strict, only in cases of suspicion of malware will an application be removed or not admitted (Google, 2013).

Hence, there are differences in admission to the application market or store between Apple and Google. However, the perception is that it is converging. As Peter Robinnet (PR) states:

---

2 The Android experience is what users experience when using a smartphone that is loaded with the Android operating system. Google strives for a consistent experience independent of the smartphone or brand that is used. The large number of OEMs makes controlling this difficult.
There is a difference but it is increasingly becoming smaller... ... Where Apple has become slightly milder, Google is clearly becoming stricter. It is converging to a balance where both methods seem more equal.”

They are increasingly going in the direction that Apple has set out earlier. Periodically Google removes a large portion of malware (Hildenbrand, 2013). To prevent new malware from coming in the Market and to assure a more consistent user experience that is free of malware, they have developed a process in which the applications are scanned for malware attributes (Rice, 2013).

Google is now screening what apps are released on the market and refuses apps sometimes. However, perhaps even more important is the fragmentation of the user experience as OEMS and operators were loading new phones with too much malware that hurt the Android experience.”

The mild policy that Google adopted in the past has led to fragmentation of the user experience on the Android platform. Many different OEMs and operators loaded their devices with Android and third party applications that could be called malware. As a result, people didn’t always realize they were using an Android operating system and the user experience was not consistent with the aim of Google. Furthermore, the malware reduced the overall quality perception the user has of the platform. Making it less attractive for clients to have their application developed for the Google platform.

The more moderate governance approach of Google is in stark contrast with Apple that has been strict from the start, leading to much criticism. As a result they made the criteria publicly available and allowed feedback from the developer:

“These criteria are still very vague and more than often apps get refused based on arguments that are unclear to the developer. Apple has become somewhat (but only somewhat) more open and explicit about what’s allowed and they’ve also added an appeals process.”

Where Apple has become slightly milder, Google is clearly becoming stricter. The admission requirements of Apple are obligatory and although they are not at Google officially, the perception of the application developer is that they are, or are at least becoming. Koen Mostert (KM):

“The path they took is in the direction of Apple who already has strict guidelines that are mandatory.”

5.1.2 Influence of developer in platform
Nevertheless, it seems that both Apple and Google have enough money to do just what they please. In particular Google has a venture capital department that invests in new start-ups through their

Malware (sometimes referred to as ‘crapware’) can be seen as software that the user did not specifically want, but is installed on the smartphone and is using resources of the smartphone. It can lead to unwanted advertisements or more serious leaking of private information of the user.
Google Venture department but is also active in acquiring new technology through their Google Acquisition department (Schijf, Hakvoort, & Truijens, 2012). It is confirmed by Robert Atkins (RA), stating:

[RA] “They [both Google and Apple] develop things and push it in the market. If they don’t have the technology they just buy the company that has got the technology.”

This method has been adopted when executing their strategy for mobile activities. They acquired Android (in 2005) and have controlled the development since then. When looking at the development of the operating system Android, it is still an open-source project. However, how it is controlled and how the developers should work on the platform is becoming increasingly more one-directional. Jurren Pen (JP):

[JP] “People can suggest features or code that might be added in an update by the Android company. Editing and approval is kept behind the screens.”

The actual editing and approval of those updates is kept proprietary and the ‘openness’ of the operating system can be discussed (Gilbertson, 2010; Goel, 2010; Vernon, 2010). The community of developers have little influence in updates to the platform, although in particular development by a community is seen as ‘open source’ (Goel, 2010). On a broader level this is confirmed by all interviewees who state that the developers have little to no influence in their relationship with the platform leader. The interaction they have in a relationship with the platform leader with respect to the development of applications is limited as well. They only do business with them when they acquire the SDK or iTunes account and when an application needs to be approved for the App store or Android Market (Android, 2013).

Nevertheless, some nuance is brought into the discussion by Peter Robinnet, Robert Atkins, and a developer at Xebia in the relationship developers have with Apple:

[PR] “Apple doesn’t work together with app developers, certainly not small ones. Perhaps the companies like Rovio (Angrybirds) or Whatsapp...”

[RA] “It [cooperation between platform leaders and developers] definitely happens with the largest brands/companies (EA, ESPN, Disney, etc...) but the partnerships are secret and/or only explicit at events like WWDC [WorldWide Developers Conference].”

[DEV-X] “For some larger developing companies I believe there are some streamlined processes. But this is not common.”

Even the larger developing companies will use the SDK and work with the rules that are set out by Apple or Google. Due to the large power distance between application developers and platform leaders, there is little collaboration. In case of problems or issues the application developers seek help from other application developers through both online as well as offline communities and events (such as Appsterdam and Cocoaheads).

5.1.3 Quality difference in platforms

It is commonly agreed that the overall quality of the Apple iPhone is higher than that of the Android systems. By vertical integration Apple has developed both operating system and phone in-house, leading to a more bug free experience of the platform.
“A reduced number of malware apps and less bugs on the platform [are seen as an image of quality]. People on Android are used to apps that crash now and then, with Apple you know it will always work.”

An application that does not have the look and feel of the Apple experience, or does not function according to guidelines, will not be accepted in the App store. Although it is claimed that the process takes up to three months, this is not common in practice. Most often in one to a few weeks the application developed by a developing company is approved and launched in the App store, whereas an individual developer can get his application approved within a day (Kokabha, 2012). The control that Apple has on its platform has improved quality but is very rigid. However, they don’t always have the qualities in-house although they think they do. This became visible when they blocked the use of Google Maps software, upon the introduction of their own software ‘Maps’. It was far from bug-free and after three months and much criticism Google was allowed to deliver its Google Maps again in the App Store (Gizmag, 2012). Nevertheless, Koen Mostert argues that this strict control of the platform is something that Google should pursue to ensure a sufficient quality is delivered.

“There are so many developers and applications, some coordination is necessary to ensure a sufficient quality level is met.”

A higher level of quality will improve the overall attractiveness of the platform. This is to the benefit of all participants as they are mutually dependent on each other within a platform. Although a developer could enter into different platforms and thereby spread the risks, the high specialization of developers can limit their ability to enter into both platforms.

5.1.4 Entry barriers for developers
Both Google and Apple have entry barriers for developers to start creating applications. Entering the Apple ecosystem has direct barriers by annual costs for a testing platform ($99) and making developing applications only possible on a Mac computer (Apple, 2013c). The SDK is actually free but without testing it’s not feasible to produce an application. For Google the direct costs can be low, as the SDK is free, but putting the application in the market means a purchase of a one-time subscription to the Android market ($25). Development of applications can be done on any computer (most often already in possession of a starting developer). However, as every OEM can sell a smartphone with Android, there are many different phones that adopt different interfaces between hard- and software. To assure operation al the entire Android platform, the application should be tested on each device. As a result, although the direct costs of development are low, in the end, the developer needs to purchase different smartphone devices and spend time testing to assure that the application will work on phones of different brands.

“That means that you need to actually own a smartphone of all those brands which can become expensive and time consuming testing it on every device.”

On both platforms, 30 % of any revenue that is created through the application goes directly to the platform leader (Kokabha, 2012). Within the App store, it is common that an application costs €0,79 or more. People are used to pay for an application, and require a certain quality level. On the Android Market, most applications are for free and people are less willing to pay for an application if the company charges any money. With Android applications different business models are used that are more complicated for a starting developer. Revenue is created through in-app advertisement, or
in-app purchases. Appropriating on the Android market is more complicated and it is experienced as an entry barrier for starting developers as observed during the interview with Koen Mostert & Jurren Pen. The developer that tries to develop an application for himself will want to appropriate on it eventually. In the case of Concapptual, they have many good ideas but haven’t started to develop them as they don’t know how to gain revenue from the application beforehand.

**[KM]** “We have enough ideas on our own that can be developed but we need some sort of launching customer to start develop it. We need to make a living somehow so simply starting developing without doing something else is not sustainable.”

This is reflected on the percentage of total applications that are in Apple’s App Store, which is 18%, in comparison with 75% with Google. However, those 18% are estimated to generate $5,1 million in revenue on a daily base, in comparison with $1,1 million on the platform of Google (Koetsier, 2013).

These numbers reduce the attractiveness of the platform for brands and companies that want an application developed. The mutual dependency between the developers and clients can reduce the attractiveness of the platform. The larger installed base doesn’t seem to compensate for the lower revenue that is generated at the moment. Nevertheless the adoption of smartphones with a Google or Apple platform is still growing (Eddy, 2013) and a new business model might be able to change the revenue that is generated per user.

In this line of reasoning it is argued that Apple is losing market share to Google as people are becoming less eager to pay for applications. They see that you can get it for free in the Android Market. As was remarked by a project manager at Xebia.

**[PM-X]** “The premium that Apple asks of their customers for the higher quality and spam-free application is not always worth the money. Although it is now more difficult to find a good application in the Android Market, this is likely to change in the future as people seem to request this.”

The difficulty in finding the proper application can be seen as a reverse salient to the Google platform. However, there are already many blogs and review sites that offer information with respect to applications such as AndroidTapp and 148Apps. These sites offer a part of the service for which Apple requires a premium. This is for instance comparison of applications or judgement of the quality that applications offer. Users often cannot see the application beforehand, they just have to pay and hope that it meets up to the demand. As a result they can be disappointed after purchase.

### 5.2 Application developers and clients

The application developer builds applications for him/herself, or for a client. When the developer builds his own application this might be done in collaboration with other developers. This is discussed in the following section. With respect to the clients; they can be internal to a company or external. In either situation, it involves a cooperation of both parties to translate the wishes of the client into a working application.

#### 5.2.1 Business relationship with a client

Most (new) projects come by referrals or mouth-to-mouth advertisement. Although advertisement is becoming more common in the market of application development. There are still no dominant
developers and especially new customers will always look within their own network for developing companies or self-employed developers.

**[PR]** “There is a dense community of application developers and if you deliver good work to certain clients this is quickly dissipated throughout the community. Furthermore it is known what specialty the individual developers have and if not, a request is made in the community that is quickly answered.”

Peter Robinnet further argued that if he gets a request from a client and he cannot do the assignment himself then he looks in the community who can, which works vice-versa. He further states that the demand for application development far exceeds the supply and the developers feel little to no competition from each other. This stimulates the trust bonds between the developers and creates a stronger social system. It resembles an organic structure in which everyone is as equal as possible that is similar to the organizations in which they operate. However, it is likely that an equilibrium in supply and demand is met in the future, accompanied with more competition among developers.

For now, the network of clients and developers is not interwoven. On the contrary, there is a gap between them as most often developers do not mingle with potential clients. This is observed at the events of Appsterdam and Cocoaheads, but also identified in an interview with the developers of Conceptual. Xebia seems to have addressed this issue by having Project managers that have more ‘business’ skills and are better capable to bridge these networks. This is an asset that the individual self-employed developer often does not have (there are few self-employed developers that are both strong on the business side as well as the development side). Larger developing firms have project managers that adopt these jobs, as relational managers or account managers. For self-employed or small developing organizations this is less common. Their resource poverty limits them in hiring these employees but they also don’t feel the need to hire them as the demand for development exceeds the supply.

For smaller clients it is difficult to judge the quality of the developer (or development company). The advertisement by word to mouth assures the client that the party that has brought the developer forward has a positive experience with him.

**[PR]** “Most often the assignment is given via friends or people you know. There is no real market where you can just hire an app developer. The assignments come around by social contact and you don’t want to abuse the relationship you have with both the client and the friend via whom you got the assignment. It is a network based on trust and reputation.”

With larger development companies reference is achieved with a list of previous performed projects. This is observed on websites of various developing companies such as Xebia or Willowtree.

**[PL]** “If you have worked for larger companies that gives a strong signalling effect to possible new clients. It shows that others have trusted you with an application that is important to them. It is proof of your skills that you can deliver a product.”
5.2.2 Developing applications for clients

The process of development involves client and developer to cooperate with each other. There are lots of knowledgeable small clients and developers. However, as Peter stated:

[PR] “...there is also a strata of clueless want-to-be application entrepreneurs.”

And this can create difficulties as only larger companies/brands have generally started to build up some knowledge about the application development process in the last few years merely by virtue of having been through it once or twice already. As a result, the developer frequently has to do more than just developing an application. In some occasions the developer will have to guide the client in a process of finding out what is possible and what the client would want.

[JP] “...you have to teach them what they want and what is possible. You guide them through the process of app development. They just have a market that they want to serve and you make everything in realizing that market.”

This requires different skills that the developer does not necessarily have. Frequently the client will only decide on functionality or design (looks) and not on how it is build or what features are possible or not. Larger developing organizations are better able in addressing these problems as they have employees with different backgrounds and skills.

Large differences exist in how the project commences and in what detail it is elaborated upon beforehand. Where small developers will merely agree on the project:

[PR] “Only a few clients require a plan of approach beforehand. Most often it is just developed in good faith that it will be developed as efficient and good as possible.”

Larger developing companies have elaborate legal documents, developed in accordance with the client. They adopt more formal approaches that are similar to those of the client.

[PM-X] “Functionality is described but not in detail, that would not be feasible. Nevertheless, the contracts are extensive in particular with larger clients that propose their own contract.”

The differences occur as the smaller organizations often lack the knowledge to develop elaborate contracts. And they also prefer spending their time on development instead of legal issues. It can be argued that this is likely to change in the near future as the market becomes more mature and both developers and clients might act opportunistically, abusing the trust of the other organization. High levels of trust together with low levels of formal control are prone to opportunistic behaviour by greedy partners (Vlaar, Van den Bosch, & Volberda, 2007). This was experienced by Marcel Groenewegen (MG) in the development of his software platform:

[MG] “The developer took the idea I had developed with the possible partnering firm and started to develop it for the other company. I trusted the developer and the partnering firm, this has clearly been damaged.”

However, all interviewees agree that the work is done based on trust, some quotes illustrate this:

[JP] “...it [trust] keeps the relationship more informal and flexible.”
“They [clients] have to trust that we will make an app as efficient as possible against a fair wage. If they do not trust us to work efficiently, then most often you work for a fixed price instead of an hourly wage. This can turn out bad for the developer as most often the planning/budget are overrun in time.”

And also the same interviewee whose trust has been abused agrees that from the perspective of the client trust is important. It can give the assurance that the developer will not act opportunistically:

“...the same interviewee whose trust has been abused agrees that from the perspective of the client trust is important. It can give the assurance that the developer will not act opportunistically:

“It is really important that you have a good connection with the developer. You must trust him to stay dedicated to your product and will not walk away with your idea. What seems complicated for you might be easy for him to make. Without the trust that he will actually do it as quick and good as possible, you will end up with a product that is expensive.”

The trust in the relationship will allow the developer the flexibility to work as efficient as possible and improves the communication between developer and client. This is important as it gives the client more control on the project and helps translate his ideas into a working product. If there is a good trust relationship between developer and client, the work might be done on an hourly wage rate basis with an agreed estimation of the delivery date. All interviewees agree that this is their preferred method of work. It is a relatively new market in which there are no pre-set prices or something alike. The work that is delivered is often tailor made and difficult to describe ex ante. As a result the client often doesn’t require the lowest price:

“They don’t require the lowest price, partly because they don’t know what the lowest price is.”

However, it is common (but to a lesser extent) to work for a fixed price per functionality. This is less preferred as stated by a developer at Xebia:

“Sometimes building the exact functionality will cause time overrun while an alternative offering the similar experience can be developed in less time... The developer can suggest something that offers similar product experience but a development time of one day [instead of two weeks].”

But with a contract that specifies the functionality beforehand, this flexibility is not present. And as the developer is paid based on the functionality or end product, he will have to put in the two weeks of time to get the result. Although the alternative would be just as good and more efficient in development. As a result developers don’t want to work with these kinds of contracts. Apart from the practical limitation in describing the functionality of the product ex ante. It is observed that developers work with an hourly wage with an estimation of the time necessary. If that time is exceeded some discount might be given.

The uniqueness of each project makes it difficult to describe contingencies in a contract ex ante. And as legal matters are often not the specialty of developers, they don’t always draft up contracts. If a contract is drafted it is not really specific. It is seen as a formal obligation to do but not much time is put into it. They are there as back-up in case things might go wrong, and are a last resort in a possible dispute. Differences are present among practitioners and it was debated if and to what extent
contracts would contribute to their business. In general they agreed that drafting up a contract should not take much time and the contract should be compact.

[PL] “I always draft up contracts, but they are there as back-up in case things might go wrong. They are a last resort in a possible dispute.”

[KM] “We don’t really make up a contract but we agree on terms such as tariff and the duration of the project. This is all agreed upon by email which is indirectly a formal agreement I believe.”

It is not their field of interest and as a result they spend little time to legal matters. They do confirm wage rates and product functionality per e-mail. It is also common to deliver a small project proposal, although the perception is that the client does not really know what to do with it. In particular smaller clients don’t always have the necessary knowledge to give substantive feedback or comments. As a result, the collaboration is based on the trust that the client will pay and the developer will live up to the assignment. Most often the assignment is given via friends or people he knows. There is no real market where you can just hire an application developer. The assignments come around by social contact and the developers don’t want to abuse the relationship they have with both the client and the friend via whom he got the assignment.

For larger developing companies it is common to draft a contract. The clients come up with program requirements that are part of the contractual agreements, sometimes in large detail.

[PM-X] “The contracts are extensive in particular with clients that propose their own contract. In the contract it is specified under what wage everyone will work. What project management method is used, under what conditions the project can be ceased or terminated, etc…”

Furthermore a description of the program that will be developed is given, including the functionalities and appearance if possible. However only the basic functionalities and look and feel. As the more in-depth functionalities are reviewed with the customer during the frequent communication moments. In general it can be said that, the larger the customer, the larger the legal documents. The contract forms can become complicated as drafting classical contracts in which the project is described in large detail is not feasible. This is due to the uncertain nature of each project.

At Xebia this is being addressed by using Agile contracts, new in the industry according to a project manager at Xebia.

[PM-X] “New in the industry are Agile contracts. They specify that a client can terminate the project within a pre-specified period (for instance 2 weeks, following Agile project management). That period is used to develop a sub-deliverable. At the delivery the client can choose to continue or terminate the project.”

With this method the client is continuously involved in the project and sees the progress that is made. Furthermore, the client can choose to cease the project if for instance the budget runs out or the project is not progressing as wished. The client will always be left with a working program but perhaps not all the functionality that he desires (Taymor, 2012). The frequent communication with
the client involves the client with the project and allows the trust between the client and developer to be improved. This is important for the developer to allow his/her creativity in the project.

The Agile contracts force frequent contact between developer and client. However, in practice even without these contract types, there is frequent contact between developer and client, although to a less extent as with Agile contracts. It is needed to communicate on the progress of the project and the development of the desired functionalities. As the developer has done more projects with a certain client, and the relationship between them has improved, this can become less frequent. Both clients as well as developers have confirmed that trust between them is important. The client needs to trust that the developer that he will work as efficient as possible.

[MG] “There are rarely any large discussions, but I believe that is a result of the long-term cooperation with the developers. They know what I want and I know what they can do, that helps avoid discussions.”

Once the project is completed, in case of the Apple platform, an iTunes account of the application is handed over to the client, or simply the actual code. With the Google platform this is similar. Pieter Lekkerkerk described that the program or code can be used as hostage for the payment, although in practice the application is handed over before the payment as the trust relationship between developer and client is good. Once the application is handed over to the client, and the client decides to distribute the application, the client enters into a relationship with the platform leader. The platform leader offers distribution channels to sell applications (App store, Android Market) and can offer marketing services or exposure to users of the platform (Jansen & Bloemendal, 2013).

However, the services that are offered to clients are standardized and the influence a client might have in a relationship with the platform leader is limited and similar to that of the developer. As a result, different organizations have entered the market that offer these services and specialize on online marketing and exposure, for instance via App Store Optimization (ASO). Seeing that the platform leaders are not addressing this market completely, they have ceased the opportunity and offer owners of applications alternatives to the platform leaders. It seems that this further stimulates the growing economy of mobile applications and the variety in species that is being developed is in support of the growing health and performance of the mobile ecosystem. It is in line with the strategy of the platform leaders to allow members of the mobile ecosystem to develop their own complementary products to the system. A virtuous cycle can be identified in which more developers attract more clients, and more users. Complementary products are being developed that support the growing mobile economy, as the organizations co-evolve towards a goal that has been set out by the platform leader.

With larger customers, development can be done on site of the client and the product is then already in possession of the client. Larger projects sometimes require down-payments, in particular with developing companies that employ several developers on a project. Developers might deliver after-service, mostly based on an hourly wage.

From the perspective of the client, a non-disclosure agreement (NDA) might be important and is sometimes used. Some ideas that the client wants developed can be easily redeployed by the developer himself. The NDA is there to prevent the developer from acting opportunistically. Or from
friends of the developer from copying the idea. However, it is argued that a NDA can give a false security as a slight alteration to the original idea will make the NDA worthless.

[MG] “The developers have to sign a Non-Disclose Agreement (NDA) to prevent them from walking away with any ideas I let them develop. Nevertheless, such a NDA is bit of a false security, the developer can discuss the idea with his friends after which his friends start to develop my idea. Although the link would be obvious, it’s not really possible to proof it in court. But the idea that it might cause trouble is enough incentive to be honest and oblige to the NDA.”

Lead-time on competitors is seen as a more valuable blocking mechanism to possible competition. Furthermore, the client needs to offer the right incentives to his developer.

[MG] “There is a continuum between dedication and opportunism, if the incentives are not right the employee will strive for his/her own goals and act opportunistically.”

In particular in the case of a long-term development project as primary product of the client this is important. The client is dependent on the progress and commitment of the developer. Developers that are seen essential to the success of the product might be given a share or an option, to assure their commitment to the project and reduce the incentive for opportunistic behaviour. This is already identified in literature as partners can become untrustworthy if the incentives are large enough (Vlaar et al., 2007).

5.3 Application developers and application developers
The application developers participate in both offline as well as online communities. They form a strong informal social system in which they help each other with technical problems as well as business problems. Furthermore, they ‘find’ each other via this social system for the joint development of applications or sometimes companies.

5.3.1 Offline events
Offline they find each other with events such as Appsterdam and Cocoaheads. These are events during which in an informal setting they share knowledge through presentations and discuss technical problems with each other. In particular Appsterdam is an event in which they focus on more than just development issues and, hence, build up personal relations. Developing companies recognize these events and participate by inviting the event at the location of the company. Some companies also organize similar activities their selves, as was observed at Xebia. By organizing frequent meetings with all employees they stimulate knowledge sharing and social bonding. During these meetings projects and innovations within the company and industry are presented and discussed. Sometimes accompanied with diner and drinks, stimulating social bonding.

It was observed that the developers do not see each other as competitors, although in fact they might be so when a client is in search for a developer. As Peter Robinnet mentions; the demand for application developers far exceeds the supply and there are little public tenders in which they could actually compete against each other. Nevertheless, in time they can be both competitors as well as collaborators at different moments.
5.3.2 Online events
These activities continue online at forums such as Stackoverflow.com and TechCrunch.com, and social media as Twitter. They know each other’s specialties and interests. And help each other with difficulties in a project or even with finding new clients.

[JP] “Many developers have experienced similar problems as you encounter and often have already written about it. There is a dense community of app developers that help each other solve problems on forums (i.e. techcrunch.com, etc...).”

By actively participating in a community of application developers, they build up their referrals among developers and share knowledge. There are online ranking systems for the skills and contribution of developers in the forums such as on the forum of Stackoverflow. An application developer with a high ranking is more interesting than one with a low ranking or inactive ranking. This stimulates the sharing of knowledge and helping others, which make the communities important contributors to the niche of application developers. Developers use their ranking as referral for new clients.

5.3.3 Collaboration of developers
Application developers work for clients but also for themselves, sometimes in a collaboration of several developers. Often it is one developer that has the idea and attracts other developer to his project.

[DEV-X] “He will have the final call on decisions and wages. They mostly work based on an hourly wage. But I have not experienced any formal contracts apart from email conversations with the agreed wage.”

If possible work is done on an hourly wage but mostly when a certain project is commenced there are no revenue’s and hence no wage can be paid at start. Work is done on goodwill or trust that once revenue’s do come in they will be distributed accordingly. Launching customers can help them in creating initial revenue’s, however, it can be difficult for a developer to find them and it is not always possible with the application that is being developed. Furthermore, the customer is likely to require a more formal form of collaboration.

In certain occasions the joint development is done in more formal collaborations such as a ‘general partnership’ (or ‘Vennootschap Onder Firma’ (VOF) in Dutch) or ‘Cooperative’ (or ‘Cooperatie’ in Dutch), although this is not a common choice.

[PM-X] “I have heard of a group of developers that recently entered into a ‘Cooperatie’. However, this is not common as they most often do not care about the legal aspects and use the method that has proven useful in the past or has received experience by some of the developers that are part of the collaboration.”

These legal forms make sharing of both costs and benefits possible and it will force the participants to ensure a certain commitment. Furthermore, the strong trust bonds between the developers give the participants an assuring feeling that both the costs and benefits are shared. For clients this is seen positive as it shows a long-term commitment to the project.
5.4 Application developers and patent holding companies

From the interviews, a different type of company was identified that plays a role among larger developing companies but also between platform leaders. These are the so called patent holding companies or ‘patent trolls’.

Patent trolls have a large asset base of patents and monitor the mobile ecosystem for infringements of those patents. If a particular patent is used in an application, that has received attention and hence has come into sight of the patent troll, then the patent troll will contact the developer and ask a fee for the use of the patent. If the developer refuses to pay it will be filled a law suit. The payment that must be made has been chosen ‘just right’ as it will not bankrupt the developer but will enforce enough costs to make the earlier project a useless investment.

[RA] “If you use technology that happens to be patented somewhere and you’re app gets successful, they will sue you. A bunch of American lawyers send you a letter in which they offer to settle for an amount that is just right. It is not enough for you to go to court and fight them (which you won’t win) and is just enough to really hurt you but you will be able to continue. In the end, you pay them and go on.”

However, not only patent trolls but also platform leaders are involved in fierce battles with respect to patent infringement in the mobile ecosystem of each platform (Puttk, 2013). The platform leaders are continuously filing for law suits, in an attempt to hinder innovations and gain licensing revenue’s for patents they have applied for. The patent trolls play a minor role within the ecosystem in comparison with these battles as they can merely hope they will make a licensing deal with a deep-pocketed developing company. However, for developers it is sometimes inevitable to use a patent owned by a licensing company, making it a matter of time until they contact you and require a licensing fee.

[RA] “Sometimes you just have to use it, such as in-game payments. It is patented by some company, but everyone uses it in their apps. You need the functionality but once your apps become successful they will contact you.”

The developers lack the resources to prevent these events from happening or actually enter into the law suit with the patent holding company. In larger developing organizations this is likely to be different as more (financial) resources are involved and there is more knowledge with respect to legal matters of intellectual property. Directive leadership of managers will try to prevent developers from using functionality that might be patented.

5.5 Summary

The case study has been executed based on interviews, observations, and is supported with secondary sources such as websites and forums. In this chapter a description of the single case study is given of the business relationships that application developers have. It is structured according to the species with who the developer is in a relationship with and thereby consists of four embedded cases.

The relationship a developer has with the platform leader involves a large power distance. The developer does not perceive to be of influence in the relationship and it seems that the platform leader can do as he pleases. The platform leader adopts a hierarchical dimension in which both
authority-based and contract-based governance mechanisms are used. In case a developer is confronted with difficulties, help is found in the on- and offline community of developers. In this community a network dimension is dominant in which the developers use trust-based governance mechanisms. To enter either one of the platforms, several entry barriers need to be taken that for a small developer can be perceived problematic. The choice to develop for a platform is influenced by the overall difference in quality between the platforms. The free character of the Google platform has resulted in a perceived lower quality of applications in comparison with Apple but a large installed base. The strict governance policy of Apple does not always result in a superior product and limitations in their capability to innovate are identified.

The relationship between developer and clients is based on a trust bond that grows stronger as the relationship evolves throughout the project. It helps the developer in expressing his/her creativity and offers the client more control on the project and budget, given the uncertain nature of the process. Often developers don’t work with elaborate contracts as they find it a waste of time and lack the knowledge in drafting them. However, larger developing companies do work with contracts and some have started using specialized types of contracts that allow both client and developer to work with the uncertainty in the process. The use of these Agile contracts can be seen as new in the industry of mobile application development. The governance dimension is that of the network dimension in which both trust-based as well as contract-based governance mechanisms are being used.

Apart from developing for clients, developers also work together in projects. They find each other through both the on- and offline communities and sometimes work together in legal forms such as a Cooperatie. The community of developers help each other with issues and stimulate the sharing and creation of knowledge. The relationships are governed with trust-based governance mechanisms in a network dimension.

Lastly, in some occasions are developers confronted with patent holding companies. Developers sometimes cannot bypass the use of patents or are simply not aware that they are using a patent. Nevertheless, the patent holding companies will scan applications for any infringement of patents and will require licensing fees if possible. The patent holding companies use contract-based governance mechanisms in a market dimension.
6 Analysis of case study
The case study has been performed from the perspective of an application developer in a mobile ecosystem. In this chapter the conceptual framework is applied to the case. In the following sections the exchange conditions and governance mechanisms are described, surrounding the relationship the developer has with organizations in the mobile ecosystem. It is broken up into the four embedded cases and is expanded with a section on the system-level implications of the business ecosystem perspective. The chapter concludes with an overview of the hypotheses that are found from the different cases, leading to a conceptual model.

6.1 Analysis of application developers and platform leader
The platform leaders in the mobile ecosystems of the embedded cases, Google and Apple, use a different approach towards the participants in the ecosystem. However, both platforms show little to none collaboration with developers or clients of developers. From the perspective of the platform leaders, they are merely users or contributors to the mobile economy they have created. In their relationships with these organizations different things are being exchanged. By standardization the platform leader has made products of the exchanged goods, being the SDK but also an entry of the developed application in either the Android market or App store.

There are large differences between platform leaders and developers. Where a developer has in general limited resources, in particular financial resources, a platform leader has a wealth of resources. The platform leaders have specific assets on which the developer is dependent. They offer software for the developers to create applications and provide in a platform in which the developer can distribute this. In return they require a part of the revenue that the developer generates. The platform is predictable and stable with limited interaction between organizations. This creates a large power distance and dependency of the developer on the platform leader.

In the following paragraphs the specie of Apple and Google is identified. Furthermore the relationship between them and the application developer is analysed. A conclusion is drawn on what influence the developer has in this relationship and what governance mechanisms are preferred.

6.1.1 The specie of Apple
Apple has developed a platform in which everything is made in-house, and if needed, new technology is acquired by taking over companies. From smartphone, to operating system, and an initial set of applications on the phone out-of-the-box. Their governance policy can be seen as strict and it gives them the control to safeguard the quality of the total user experience. In return, it allows them to ask a premium for the majority of applications that are sold in the App Store.

The application economy is a relatively immature market as it finds its origin in 2007. Apple has adopted, from the start, a strict governance policy with respect to the application developers and owners of applications. However, where the admission criteria used to be unknown, they have specified this in a later stadium, making it more transparent. Furthermore, the event of iOS 6 Maps is a sign of their slightly less strict governance policy of recent times, reducing their ability to coordinate the ecosystem. They have tried to develop this software in-house but delivered an application that did not meet up to the demand. Continuous vertical integration is susceptible for incumbent inertia and merely incremental innovations (Tiwana et al., 2010). Their corporate venturing activities are there to prevent these concepts from happening but clearly this is no
guarantee. The many criticism forced them to make an exception to their policy and allow an alternative from main competitor Google. It shows how Apple is forced to become more flexible and adapt to their environment.

Apple offers the developers a stable and predictable set of assets in the form of a platform and complements. A clear characteristic of a keystone (Iansiti & Levien, 2002). However, they adopt a strict governance policy with respect to the development of applications and try to vertically integrate as much as activities as possible. The connections with developers and possible clients are supply-chain like and by controlling the App Store they control a part of the resources that flow through it. They keep their technology in-house and close it off to external organizations. This limits the spawning of innovations from other organizations in the ecosystem and allows them to exploit the ecosystem (Williamson, 1994). The rate of exploitation is, however, chosen in a fashion that it leaves enough revenue for developers to keep it interesting to participate. As already identified in literature (Kokabha, 2012), the empirical results show that Apple can be seen as a dominator in the mobile ecosystem. Predominantly a physical dominator but with their control over the complementary products and their deduction of value from it, they are to a certain extent value dominators.

6.1.2 The specie of Google
Google has developed a platform based on an open-source operating system and joint development of a smartphone. The initial Android Market was free to enter for everyone and in combination with the smartphone made a transparent platform. It attracted many users and developers, but also malware and dubious organizations. They were forced to adopt guidelines and a system to scan for malicious software and remove it from the Android Market.

In recent years however, the development of technologies needed for the platform have increasingly been developed in-house. If the technology was not developed in-house, it was sought for in the market via corporate venturing. They orchestrate the development of applications by guidelines and use the large power distance with developers to force them to oblige. A similar share of the revenue (as in the Apple ecosystem) generated through the sales or in-app sales of applications on the Android market is extracted from the developers. And although the Android operating system is open-source, they maintain control over the software. From the perspective of the developer, Google could be seen as keystone organization in the past. However, the large power distance that has been created and the proprietary technology and guidelines that have been developed make Google a dominator for developers in present times.

It can be argued that this evolvement is inevitable as maintaining trust relationship with organizations in the ecosystem is labour intensive and thereby costly. The app economy is becoming more mature and is shifting from an exploration phase towards an exploitation phase. The rules and procedures have been standardized to offer a predictable set of assets for the organizations involved with the development of complementary products and services (Ghazawneh & Henfridsson, 2013). Maintaining a keystone strategy can become difficult as shareholders require profit maximization. Their elaborate venturing activities help them in remaining innovative and prevent mere incremental innovations and incumbent inertia.
6.1.3 Relationship between application developer and platform leader

The success of a platform is highly dependent on the availability of applications that are being used by the end users. To this end, the relationship between developer and the platform leader starts with the acquisition of the SDK. Although there is a price attached to development in the platforms of Apple and Google, there are no real alternatives as they together have a market share of approximately 90%. By standardizing the interfaces between developer and platform leader, they keep the development of the core product and the complementary products strictly separated, and offer them a stable and predictable set of assets to develop the complementary products and services.

The platform leaders seem to adopt a strict governance method in the ecosystem that doesn’t allow that much influence of the developers or clients in the platform. This can limit the spawning of new innovations by the developers, and the growth potential of the platform (Jansen & Cusumano, 2012). The success of the platforms seems to contradict with literature on platform strategies as both Google and Apple are successful. The role that the platform leaders adopt is in providing in the infrastructure with rules and procedures to use. It is a translation of their strategy and an implementation of the shared goal they have set out. It is being adopted by the participants in the mobile ecosystem without much argue.

However, collective action does seem to influence the platform leader as happened with the Maps application of Apple. Organizing that collective action seems to spawn organically from within the mobile ecosystem but only after a certain threshold has been reached. The threshold seems to be determined by the power distance that developers/clients as well as users feel, but also by the ability to act with the policy or procedures that are given out by the platform leader. It is common that participants of the ecosystem seek help from each other when they are not able to solve an issue. But once the community is not able to find a solution, they turn towards the platform leader in await of a response. Within the platform of Google this is possible by making development requests for the Android operating system. With Apple, however, this is more difficult as it is developed in-house.

The platform leaders maintain control over the development of the core product in a hierarchical dimension, internal to the firm. This allows them to maintain the high asset specificity and use relational mechanisms within their organization. It is necessary to maintain some form of control in order to appropriate economic rents from the platform (Eaton et al., 2011). Within the ecosystem they govern the development of complementary products by use of the power distance between developers and dominator and standardized interfaces such as SDK and guidelines. It is a top-down approach of governance and with contracts and guidelines they safeguard the quality of the ecosystem for the actual users. The SDK can be seen as a tangible asset that is governed through formal control in the market dimension. Although the developers have a choice to oblige to the guidelines, the large power distance forces them to adhere with an authority mechanism. There is a tension between developer and platform leader in which the developers aims to be independent and the platform leader tries to exert control (Tiwana et al., 2010). It is also a paradoxical tensions as the control that is exerted by the platform leader can limit the ability of the developers to develop innovations, although it can be exactly the new innovations that require more control (Eaton et al., 2011).
When an application is developed and is ready to be published, the developers or clients (depending on the owner of the application) agree to give 30% of the revenue to the dominator. In return, their application is available in the App Store or Android Market. The project properties are clear beforehand as it is simply an application that generates revenue. There is no uncertainty in the outcome as it will either generate revenue or will not. A contract is common for this type of agreement in which the project properties are predictable and can be verified afterwards (Williamson, 1979). There are little to no environmental changes that require alterations to the contract. Furthermore, the technological uncertainty is reduced by publishing an update for the Operating System, and the guidelines involved, well before the actual release for end-users. This gives the developers the time needed to get acquainted with the update and technology. Although the transaction can be reoccurring, there is no need to renegotiation due to the large power distance. It is argued that only in certain occasions a developing company with many customers might have a streamlined admission process, by which the time to market is reduced for their clients. Furthermore, the complexity of tasks is not an issue as the dominator is under the assumption that the developer has the necessary skill. Summarizing, these exchange conditions fit the criteria for a contract-based governance mechanism in a market dimension that is supported by authority-based governance mechanisms.

The platform leaders adopt similar governance mechanisms to control their platforms. Furthermore, entry barriers exist for both platforms. In that sense a developer will not choose a platform based upon the financial incentives but on the market that he/she can reach or the feedback that is available when solving issues. As Tiwana et. al (2010) describe, if a platform leader can make it more interesting for the developer to participate in their platform, the developers might switch to the other platform. This is described as ‘tipping’ in which rival platforms pull away developers from each other. As the platforms are in a multi-sided market, more developers will attract more users, increasing the installed base and market share of the platform. To prevent tipping, platform leaders would be better off by creating loyalty among the development community towards the platform leader. This can be achieved by creating a feeling of goodwill or improving loyalty (Rus & Iglic, 2005), and this aspect seems to be neglected by the platform leaders at current.

Furthermore, developers have reacted to the strict control mechanisms of the App store of Apple by circumventing it. By developing unofficial installers, developers were able to launch and run applications without control of Apple. This so-called ‘jail breaking’ of the iPhone eventually allowed users to download and run applications from portals other than the official App store (Ghazawneh & Henfridsson, 2013). The strict governance mechanisms have led developers to act opportunistically and find a method that is in their perception better aligned with their incentives towards participation in the platform.

6.1.4 Summary
Both Google and Apple can be seen as a dominator from the perspective of an application developer in the mobile ecosystem. They are predominantly physical dominators but try to exploit the ecosystem by deducting value from the sales of complementary products. The complementary products are developed and sold by application developers on the App Store or Android Market.

The relationship between the application developer and the platform leader is surrounded with a large power distance. This gives the application developer little to no influence in the relationship.
The governance mechanisms that are used are chosen by the platform leader. The platform leader will prefer the use of authority-based and contract-based governance mechanisms. Allowing the platform leader to control, safeguard, and coordinate the development of complementary products in its mobile ecosystem from a top-down approach.

The analysis of the relationship between developer and the platform leader has led to the following hypothesis.

\[ H1: \] In a relationship between an application developer and a platform leader, contract-based or authority-based governance mechanisms will be used, rather than trust-based governance mechanisms.

### 6.2 Analysis of application developers and clients

The application developers and clients live in a so-called app-economy. It is a market of complementary products to the mobile ecosystem of either Google or Apple. The client of an application developer (or developing company) is a company that either wants an application as core product or as support for their core product. What is being exchanged is the process of development of an application as a service and the actual application as a product. They can outsource the activity to an external developer. Due to the specialism of mobile application developers and the newness of the market, an internal mobile application development department is not likely to exist within large organizations/brands that want an application developed.

Whereas the relationship with the platform leaders has a large power distance, this is much smaller between clients and application developers. Still, there are large differences between the companies, in particular with respect to resources they have. The difference in capabilities and assets makes the client outsource the activity. Both developer and client frequently fall in the category of SME (with a large portion of self-employed developers), and have similar characteristics and preferences in their relationship. However, once the application is developed, the client enters into a relationship with the platform leader as the application is publicized in the market or app store.

The strict policy of the platform leader as identified in the relationship with the developer, is in line with the differences between SMEs and large enterprises. Communicating and working within and between SMEs seems to be based on interpersonal (and inter-institutional) trust whereas the platform leader will base as much as possible on formalized contracts, rules, and procedures. The differences exist with both clients and developers, in relationship with the platform leader. And the difference in their preferences can cause struggles in the relationship. The strong power position of the platform leader will make them decide what governance mechanisms are used and forces the SME to adapt to the situation. In this line of reasoning it is expected the business relationship between platform leader and client will be similar to that of the platform leader and developer. Both contract and authority based governance mechanisms will be used rather than trust-based governance. And although the platform leader is actually gaining revenue from the client, the power distance seems to be equal to the relationship with the developer.

In the following paragraph the relationship the developer has with clients is described. It is followed by a conclusion on the preferred governance mechanisms for the developer to influence the relationship.
6.2.1 Development by external organization to a client
The client that has decided to outsource the application development to an external organization is likely to look in his own informal social network for a suitable candidate. Most often the client is inexperienced in the development of mobile applications. Therefore he will seek a developer that has references to be trustworthy, good, and honest. This is done as safeguarding mechanism to prevent opportunistic behaviour by the developer. Mouth-to-mouth referrals are then appealing as there is a certain commitment by the developer to the client and his reputation is at stake if the client becomes disappointed.

The developer offers services based on intellectual capital and tacit knowledge. Due to the immature market, the client often has little experience in the service of development. This creates a large asset specificity for the developer in relationship to the client. Furthermore, the process of development is often uncertain and difficult to specify beforehand (Taymor, 2012). This creates uncertainty in the technology and the behaviour of the organizations. Furthermore, the process of development requires creativity in finding solutions to new problems and in the required functionality of the application. The developer will experience difficulties in developing exact described functionalities. However, sometimes the developer can suggest alternatives that offer similar user experiences but in less time developed. It is a continuous innovation project that needs to be flexible upon changing environments. Lastly, the complexity of tasks is high and in some occasions under time pressure to deliver it before a deadline.

These exchange conditions surrounding the development process make authority-based and contract-based forms of governance less effective and in particular trust-based governance mechanisms more effective. Both the developer as well as the client will prefer a relationship in which there is trust that the other organization will deliver what is agreed upon and not act opportunistically. The mechanisms of trust and reputation are then important drivers.

Communication between developer and client is important if a trust relationship is to develop. However, often developers lack the business skills necessary to communicate on the same level as a client would want. It can be surmised that this limits self-employed developers in becoming larger companies with employees. Developing companies have account managers and project managers that deal with these activities, thereby removing the reverse salient that hinders the self-employed.

Drafting a contract that describes exactly what the process will involve and that incorporates all contingencies is difficult and not feasible with application development (Hoetker & Mellewigt, 2009). Furthermore, application developers believe it be time consuming and costly. Nevertheless, a contract of some sort is preferable to at least describe on what terms the work is done. Some larger developing organizations adopt new contract methods such as Agile contracts, although it doesn’t find large practice yet. Smaller developers have not seemed to adopt this. Often they lack the skills to draft up legal documents and are not interested in it. They are more concerned with building up a good reputation and might use the application as hostage in case a client acts opportunistically. Nevertheless, the agreement that the developer makes with the client is either verbally (e-mail) or spoken, which is a legal agreement and a clear example of a contract-based governance mechanism. Determining price is based on experience and complexity of the project and is also a governance mechanism in a market dimension. Although the primary governance mechanism is a contract, they
are still largely dependent on the trust that both organizations will not act opportunistically. It is a clear example of the simultaneous use of different governance mechanisms.

However, although using both trust-based as well as contract-based governance mechanisms are preferred, using them simultaneously can be redundant and thereby costly. Furthermore, the use of formal control mechanisms can be perceived as a signal of low trust. In that sense, contract-based mechanisms can damage the quality of the relationship (Pittino & Mazzurana, 2013). Also, as the developer takes the relationship for granted, he might be blind for opportunities that pass by. Too much trust can cause naiveté and create a take-it-for-granted-mindset (Vlaar et al., 2007). Some opportunistic behaviour is therefore essential and also a pre-condition for transaction cost economics.

It can be concluded that a balance is sought after in which the positive safeguarding and control effects of contract-based governance complement the coordinating and control effects of a trust-based mechanism.

6.2.2 Summary

An organization that wants an application developed is likely to outsource these activities to an external developing organization, and will become a client of the developer. Finding a developer by mouth-to-mouth advertisement makes use of the reputation effect the developer might have. The developer will give a certain commitment from the start, which the client can use as safeguarding mechanism to prevent opportunistic behaviour. The exchange conditions surrounding the development are that of high asset specificity with intellectual capital, and uncertainty in the technology and behaviour of the organizations. Furthermore the complexity of the tasks is high and the work needs to be flexible to adapt to changing environments. Contracts are not always used but an agreement on the terms and conditions is made either verbally or spoken.

The governance mechanism of trust is more effective in these circumstances and is the preferred mechanism of both organizations. Furthermore, to a lesser extent is a contract and price mechanism used from the market dimension. Using these mechanisms will give the client a safeguarding, controlling, and coordinating mechanism due to a price mechanism and reputation mechanism. Furthermore, the trust relationship will prevent opportunistic behaviour of the developer. The developer can use the application as hostage in order to receive the payment, and uses a trust mechanism to assure the client that his solutions are most suitable. His reputation can be used for the client to choose him.

The analysis of the relationship between developer and clients has led to the following hypothesis.

H2: In a relationship between an application developer and a client, trust-based or contract-based governance mechanisms will be used, rather than authority-based governance mechanisms

6.3 Analysis of application developers and application developers

An application developer can work on different types of projects. It is common that they work for clients, as well as on own ideas and projects. It is also common that the latter type of projects is developed with other application developers. What is being exchanged depends on the context of the relationship but can be either resources or the development of an application. This is line with
their characteristics as an SME. They have limited resources and therefore look in their network to acquire these resources. Furthermore, the high interpersonal trust that exists within the organization is reflected in their preference for trust relationships on an inter-organizational level.

The application developers gather both in offline as well as online events and forums. Together they form a community in which they share resources, such as knowledge and information with respect to clients. The social relationships from these communities are sometimes used to start a project, either in a legal collaboration form or not.

In the following paragraphs the exchange conditions surrounding the relationships within the community are discussed, followed by the relationship between developers in a joint development project. The section concludes with the preferred governance mechanisms for the developer in relationship to a community and other application developers.

6.3.1 Relationships in community

Developing applications involves knowing certain programming languages. Often the interests in programming have evolved over the years and they have gained their knowledge from school books as well as online forums. There are ranking systems on these forums with the scores of contributions by members. The more good contributions a member has, the higher the ranking. It has a strong reputational effect that inspires other members to share knowledge and help others. This in return will give them a higher score on the ranking. It is a virtuous cycle that helps all members involved in the community. Members with a high ranking have a better reputation than members with low ranking and this can creates power distances between members. Nevertheless, the informal character of the network limits the abuse of the power position.

Sharing resources in the community has let to strong informal social connections. The members trust each other and have built up goodwill, and reputation. This is reinforced by the homogeneity of the community, which reinforces the trust between the members (Powell, 1990). Everyone’s interest is that of development which involves high asset specificity with a shared passion. The interconnectedness of the network and flexibility to adapt to new environments is a breeding ground for spawning of innovations (Hoffmann, 2007). However, it limits them in commercializing the ideas and projects as they often have limited business skills.

As someone builds an interest in application development, sooner or later he will get acquainted with the online forums. Either through mouth-to-mouth advertisement or when searching for solutions to problems. The community has a low barrier of entry to new comers and makes it a welcome place to come by. Once more familiar within the community other members might hand over projects that they cannot do their selves. Certain developers on the forums act as bridge players. They connect clients with developers or developers with other developers. In return they get goodwill and reputation, and sometimes a share of profit from a project.

Similar conditions are present during offline events such as Cocoaheads and Appsterdam. The events are there to share knowledge en see the faces behind the online names. By meeting offline as well as online friendships evolve. The high interpersonal trust between the members help mitigate the risk of opportunistic behaviour through their frequent interaction.
The governance mechanisms in a network dimension are the sole mechanisms at work in the relationship between developers. Trust, reputation, goodwill are all important as the relationship becomes stronger over time. The mechanisms prevent opportunistic behaviour and help build up a culture of sharing resources, and helping others.

Their preferred mechanism in a relationship within the community seems to reflect in their relationship with clients. It is easier for them to replicate these arrangements at the inter-organizational level with clients (Pittino & Mazzurana, 2013). As the developer and client have frequent contact to tune the functionality of the application towards the desired result, the trust bonds grow which is to the benefit of a project.

6.3.2 Relationships in collaboration
Developers work for clients, but also for their selves or with other developers. In the latter situation they often meet at online and offline events and already have a strong social connection before they enter into a collaboration. If they don’t already know each other in person they can use the ranking systems from online forums, or referrals from other developers.

In time, the developers can be both competitors as well as collaborators in developing applications. Although for now the demand exceeds the supply, an equilibrium is likely to emerge in the future. This will stimulate competition and can change the dynamics in the ecosystem. The openness of the community is expected to reduce and the business relationships are likely to become more formal. A developer that is able to respond swiftly to the changing environment is assumed to prosper in the ecosystem.

A developer that has an idea for a project gathers developers around him that he deems necessary for the successful execution of the project. The exchange conditions as with clients are similar, however, the preferred governance mechanism, those of the network dimension, seems stronger. The type of collaboration is often via an agreed hourly wage but in some situations a legal form is more appropriate. Again, the developers have difficulties with this, and as a result it is not that common. However, in comparison with a relationship with clients, a hierarchical mechanism of authority is used during collaboration. The founder or originator of the idea often has the final call in decisions relating to the project.

Within this type of collaboration the governance mechanisms from the network dimension find most practice. The relationships are often existing and build on trust, goodwill, and reputation. A contract-based mechanism is used for agreeing on wage, terms, and conditions. However, in comparison with a relationship with clients, a hierarchical mechanism is used as the developers enter into a form of collaboration, either in a legal contract or not.

However, although trust-based governance is the primary mechanism, other mechanisms also find practice but to a lesser extent. Purely collaborating based on trust is susceptible to opportunistic behaviour (Vlaar et al., 2007), and even a small safeguarding or controlling mechanism can help prevent this. Again a balance is sought after in which contract-based and authority-based governance mechanisms support the trust-based mechanism in the network dimension. It is in line with research by Poppo & Zenger (2002) that identify that trust and formal control act as complements.
6.3.3 Summary
Developers participate in communities that meet each other online and offline. They build up an informal social system of relationships in which developers help each other with problems and projects. Ranking systems on the online forums create virtuous cycles, creating a culture in which it is good to share resources. The governance mechanisms of the network dimensions are used between the developers in the communities. The mechanisms prevent the occurrence of opportunistic behaviour.

The collaboration projects of developers often originate from the communities. As the predominant mechanism in the communities is trust, this finds practice in these collaborations as well. Nevertheless, to a small extent contract-based mechanisms are used to agree on the terms and conditions of the project. And the final decision is often made by the originator of the idea, in which an authority based mechanism is used.

The analysis of the relationship between developer and other developers has led to the following hypothesis.

**H3:** In a relationship between an application developer and other application developers, trust-based governance mechanisms will be used, rather than contract-based or authority-based governance mechanisms.

6.4 Analysis of application developers and patent holding companies
The relationship a developer has with a patent holding company is limited. It does not find practice often but from the perspective of a developer this can be seen as positive. If a patent holding company does seek contact with a developer, a contract-based mechanism is used. What is being exchanged is the use of a patented technology. The developer is required to pay licensing fee for the patent and is confronted with a lawsuit if not obliged with the settlement.

From the business ecosystem perspective the specie that this company has taken can be seen as a parasite. Its sole purpose is to deduct value from contributors in the ecosystem and gain revenue from the patents they have acquired over time. They find a balance in the fee they can ask, making it just affordable for the developer to survive. They do not contribute to the ecosystem in a positive manner.

The exchange condition involves high asset specificity in the advantage of the patent holding company. This creates a large power distance as the developer often has little knowledge on legal aspects. It is purely based on patents and price and only a contract-based mechanism is used to control and safeguard the patents. However, given the lack of knowledge on legal aspects the developer often has no influence in the relationship. Exerting influence would require hiring legal counsel, which in turn is likely to reduce any profits.

The developer will have little to no influence in this relationship, apart from preventing the relationship to exist. In this line of reasoning it can be argued that more formal control mechanisms should be used during development of applications to prevent the use of patented functionalities. Once in the relationship, the costs of influencing the formal control mechanisms exerted by the patent holding company exceed any profits. As a result the developer is likely to have no other choice than to accept the licensing fees.
Lastly, the relationship the developers have with patent holding companies have led to the following hypothesis.

**H4:** In a relationship between an application developer and a patent holding company, a contract-based governance mechanism will be used, rather than a trust-based or authority-based governance mechanism.

### 6.5 System level analysis

The analysis that has been performed thus far has primarily focused on the dyadic relationship that the developer has with the different species in the mobile ecosystem. From the interviews that have been conducted it became clear that the interviewees were not aware of the concept of a mobile ecosystem, neither of governance. However, from the perspective of the mobile ecosystem these organizations interact and have, among other aspects, a dependency on each other as they co-evolve towards a shared goal. In this section a system level analysis is given incorporating aspects of a business ecosystem and their effect on the hypotheses that have been developed.

#### 6.5.1 Mobile ecosystem analysis

There is a mutual dependency between platform leaders and the organizations involved in the mobile ecosystem. The platform leaders need the clients to offer business opportunities to developers, and attractive applications to users. In a similar fashion, the developers need the platform leader to make a stable operating system that is bug free and offers unique possibilities for applications. The clients don’t have the capabilities to develop mobile applications but have a market or idea that can be exploited. To achieve this they have to enter into a relationship with the platform leader but can only do this with use of an application developer. The mobile ecosystem can be described as a multi-sided market in which the mutual dependence makes the organizations co-evolve as they work towards achieving a shared goal.

The lack of competition in the ecosystem among developers, limits the dynamics at the system level (Asseldonk, Berger, & Hartigh, 2001). Organizations are not by means of competition forced to continuously renew themselves and limit their abilities to do exploration. However, the platform leaders try to govern the developers towards innovation by constantly updating the platform. These updates are in the form of new possibilities with the platform in the combination of hard- and software. Although the developers perceive a low feeling of competition, without staying updated with the continuous changes, they can quickly become outdated. The speed at which the environment changes is very high and the ability of the developer to be flexible and adapt to the changing environment is essential. However, until the supply equals the demand, the developer will not be left out of business that easily.

Nevertheless, developing organizations can ‘die’ in the ecosystem and new organizations are ‘born’. There are some entry barriers to take but the rate of renewal by new entrants is high. The app economy is becoming larger every year and it seems that everyone wants to have a piece of the pie. The market that brands or companies can reach with mobile applications is global and the increasing smartphone adoption rate is promising for all involved.

#### 6.5.2 Influence on hypotheses

The hypotheses that have resulted from the preceding analysis are influenced by the context in which it is embedded. A virtuous cycle is identified by the mutual dependency upon each other. It
causes the ecosystem to evolve and thereby change the context of the mobile ecosystem. The market is becoming more mature, and thereby the hypotheses might change as formal control mechanisms become more important. This is particular relevant for relationships were contract-based and authority-based governance mechanisms are not the dominant mechanisms in the relationship. The relationship between developers and clients, and between developers and other developers are likely to be influenced by this. However, trust-based governance mechanisms can substitute formal control mechanisms such as contract-based governance mechanisms.

As the technological trajectory of the platform evolves the platform leader will adapt its strategy to the reaction of the organizations influenced by the strategy. As analysed, the current governance mechanisms can limit the ability to produce radical innovations and thereby limit the growth potential of the ecosystem. A solution to the inability to innovate can preside in adopting a more open approach towards the organizations in the ecosystem that develop complementary products and services (Jansen & Cusumano, 2012). This would require the platform leaders to adopt a keystone strategy instead of a dominator strategy, in which trust-based governance mechanisms become more important.

6.5.3 Summary
From the perspective of the mobile ecosystem the organizations interact and have a mutual dependency on each other as they co-evolve towards a shared goal. The platform leaders need the clients to offer business opportunities to developers, and attractive applications to users. In a similar fashion, the developers need the platform leader to make a stable operating system that is bug free and offers unique possibilities for applications.

A virtuous cycle is identified by the mutual dependency upon each other. It causes the ecosystem to evolve and thereby change the context of the mobile ecosystem. The market is becoming more mature, and thereby the hypotheses might change as formal control mechanisms become more important. Furthermore, the current governance mechanisms can limit the ability to produce radical innovations and thereby limit the growth potential of the ecosystem. The platform leaders can adopt a keystone strategy instead of a dominator strategy, thereby stimulating exploration instead of exploitation and stimulating the development of radical innovations. In such a relationship trust-based governance mechanisms will become more efficient and effective.

The balance of governance mechanisms in the relationship between the organizations in the mobile ecosystem is likely to change, and thereby can change the hypotheses. Trust-based governance mechanisms are more effective and efficient to coordinate and control the exchanges. Furthermore, contract-based governance mechanisms will see more practice to control and safeguard the exchanges. Due to the speculative nature of these findings, the influence on hypotheses is not translated into new hypotheses.

6.6 Hypotheses
From the case study and analysis, the several hypotheses have been formed from the perspective of the application developer. They form the primary results of this research and are subject to future research. They are listed in this section and visualized in the conceptual model, Figure 4. It is based upon the conceptual framework from the literature analysis.
6.6.1 Relationship with platform leader  
**H1:** In a relationship between an application developer and a platform leader, contract-based or authority-based governance mechanisms will be used, rather than trust-based governance mechanisms.

6.6.2 Relationship with clients  
**H2:** In a relationship between an application developer and a client, trust-based or contract-based governance mechanisms will be used, rather than authority-based governance mechanisms.

6.6.3 Relationship between application developers  
**H3:** In a relationship between an application developer and other application developers, trust-based governance mechanisms will be used, rather than contract-based or authority-based governance mechanisms.

6.6.4 Relationship with patent holding company  
**H4:** In a relationship between an application developer and a patent holding company, a contract-based governance mechanism will be used, rather than a trust-based or authority-based governance mechanism.

![Figure 4: Conceptual model following the hypotheses showing directional difference hypotheses.](image-url)
7 Discussion and conclusions
This study has researched if and how application developers can influence their relationship with other organizations in a mobile ecosystem with the aim to derive testable hypotheses. The findings, limitations, and recommendations are given in this chapter. A reflection on the process and approach concludes the thesis.

7.1 Main findings and contributions
The findings of this research can be summarized with the answers to the sub-questions, leading to the research question. The hypotheses are the main findings and are subject to future research. The contributions to theory conclude this section.

7.1.1 Sub-research questions
A set of sub-questions are designed to help find an answer to the research question of this study. The sub-questions and answers are as followed:

**SQ-1:** What definition of governance by an application developer in a mobile ecosystem can be derived from literature?

The literature study resulted in a definition of governance from the perspective of a mobile ecosystem and a mobile application developer. Following Jones et al. (1997) p.914, governance can be seen as adapting to environmental contingencies, and coordinating and safeguarding exchanges in relationship with a select, persistent, and structured set of autonomous firms (as well as non-profit agencies) engaged in creating products or services based on implicit and open-ended contracts. With this definition it is the relationship that an organization tries to influence. Influence can be defined as safeguarding, controlling, and coordinating exchanges. This is different from governance by a platform leader in which the leader will try to incentivize collective action through governance mechanisms. In that sense, the application developer will try to influence the direct relationship it has with other organizations in the business network from a bottom up approach. However, as the developer is part of a mobile ecosystem, there are system level dynamics that can influence the exchange conditions and mechanisms in the relationship.

**SQ-2:** What governance mechanisms can be identified from literature?

From transaction cost economics theory, three dimensions (or institutions) are identified. In the network dimension, the dominant governance mechanisms are trust, reputation, and goodwill. In the hierarchy dimension, the dominant mechanism is authority, which in an inter-organizational setting is a distance of power between organizations. In the market dimension the dominant mechanisms are contracts and prices. However, governance mechanisms are non-mutually exclusive and can be used at the same time. If for instance a contract is drafted, the organizations in the contract trust each other to live up to the obligations.

**SQ-3:** What exchange conditions influence the governance mechanisms that can be used?

The exchange conditions surrounding a transaction influence the governance mechanism that is more, or less, useful. The exchange conditions are; asset specificity, behavioural and environmental uncertainty, frequency of transaction, and task complexity. The exchange conditions are influenced by the product or service that is transacted, and the organizations that are involved in the
transaction. The content of the product, or service, is primarily the development of applications but can also be admission to the market or licensing of patents. Furthermore it is assumed that the organizations in the transaction are rationally bounded and at least some are opportunist. This can be argued however, as it is perceived that developers are not very opportunist. They share resources such as knowledge and new clients, and have relationships that are strongly based on trust and goodwill. It can be surmised that the lack opportunistic behaviour is due to the low feeling of competition among the developers. Something which can change in the near future as an equilibrium in supply and demand will be reached, stimulating the incentive for opportunistic behaviour.

**SQ-4:** What governance mechanisms can be identified in practice?

Developers are rarely aware with the concept of business ecosystems but have sometimes heard of mobile ecosystems. However, they don’t know what governance entails or how it can be used to influence a relationship they have with other organizations. After operationalizing the concepts in questions in the interview, it seems that they do employ governance mechanisms, although they are not conscious of it. By far the preferred governance mechanism of application developers is trust-based. As far as the exchange conditions allow it, they use it in all their relationships with other organizations in the ecosystem. To a lesser extent contracts are used to agree on terms and conditions.

**SQ-5:** How do governance mechanisms differ according to the relationships with different types of species in the mobile ecosystem?

The differences in governance mechanisms that are used in a relationship between a developer and species in the ecosystem have led to the hypotheses that are elaborated upon in the next section. From the results in the case study, a difference in governance mechanisms in a relationship with a platform leader could not be identified. Both Google as well as Apple are seen as dominators from the perspective of an application developer. There is too much power distance between developer and platform leader for the developer to be of influence in the relationship. The problems that developers face are solved with help of the community in which the developers participate in, further reducing any interaction with the platform leader.

### 7.1.2 Hypotheses

The differences in governance mechanisms that are exerted dependent on the exchange conditions which are influenced by the type of organization with who the developer is in a transaction. It has led to the development of several hypotheses. As described, what is being exchanged is primarily the development of applications, but can also be patents or distribution of applications. The exchange conditions for the development of applications involve that of high asset specificity and high technological uncertainty. Furthermore, the complexity of the tasks is high as it involves intellectual capital intensive processes. These conditions can create an information asymmetry between the organizations.

In the relationship an application developer has with a platform leader, the developer will have little influence. What is being exchanged is the admission of an application to the market or store, and revenue that is generated through the sales of, or through use of, the application. The platform leaders Google and Apple are both identified as dominators. In contrast with research that has
identified Google as keystone organization (Kokabha, 2012). The difference arises with the perspective that is chosen in the ecosystem. From the perspective of a developer both Google and Apple are seen as dominators that seem to give no support apart from general rules and guidelines in the form of a SDK. The developer has a dependency on the platform leader as it decides on the admission of applications, creating a large power distance between the two in the advantage of the platform leader. The platform leader will determine what mechanisms are used in the relationship and the developer can only adhere and assure that he is aware of what is used. The frequency of the transaction of admission of an application is extremely high as every developer engages in the same relationship. Everything is therefore standardized, limiting renegotiation costs and reducing uncertainty (Williamson, 1979). Following Williamson (1979) an authority based mechanism would be preferred but as the platform leader is not planning to integrate these activities within its organization, a contract-based mechanism is used to assure the developer shares the revenue accordingly as alternative to an authority-based mechanism.

For the development process the platform leader has developed guidelines that the developer is obliged to follow (either implicit or explicit). The guidelines are codified and there are no difficulties in describing a contract ex-ante. Again a contract-based mechanism is used, in line with previous research (Vlaar et al., 2007; Woolthuis, Hillebrand, & Nooteboom, 2005). However, in the market dimension, also an authority mechanism is used by telling the developer how the actual application should look and feel, based on the high asset specificity of the dominator with respect to the platform.

However, the platform leaders have a strong dependency on the developers on a system level. The mobile applications are key to the success of the platforms and without developers users are not attracted to the platform. This is evident from the small market share that for instance Blackberry and Windows mobile have. A platform leader that is able to create an environment in which there is more interaction with the developer is likely to thrive as the platform leader has knowledge on how to improve both applications and distribution of applications. It seems that their current strategy is to let the participants of the business network develop all the necessary types of organizations. On the long-term, however, this can impair their ability to appropriate from the platform as for instance other application markets become successful and surpass their primary revenue generator. Involvement of the platform leader seems essential and a balance must be sought for in the costs associated with involvement and the improvements in quality of applications and market by the involvement.

**H1:** In a relationship between an application developer and a platform leader, contract-based or authority-based governance mechanisms will be used, rather than trust-based governance mechanisms.

The relationship an application developer has with its clients has much less distance in power. There is a more homogenous understanding between them as the incentives are better aligned. What is being exchanged is the development of an application. The exchange conditions for the development process create an information asymmetry in the advantage of the developer. However, the developer will not gain revenue without the client and is likely to cooperate, balancing the distance in power the information asymmetry might have created. The development process involves high technological uncertainty and high asset specificity. The process is difficult to describe ex ante.
although the functionality of the application is not difficult to describe in advance. The uncertainty in the process requires flexibility in the relationship to adapt to the environment. A trust-based governance mechanism offers the required flexibility in the relationship (Geyskens et al., 2006).

However, it involves an increased risk of opportunistic behaviour by either organization. Self-employed developers are less likely to adopt contract-based governance mechanisms as they often lack the required knowledge and see it as a formal obligation. Nevertheless, developing companies with several (or more) people employed, and larger clients, do prefer contract-based governance mechanisms as support in the network dimension of the relationship. It is surmised that the lack of use of contract-based mechanisms by self-employed developers is due to the immature market and is likely to change as an equilibrium in supply and demand is reached. This will create more competition and stimulate the incentive for opportunistic behaviour. It will require more safeguarding and control of the relationship and therefore stimulate the use of contract-based governance mechanisms (Jones et al., 1997).

**H2:** In a relationship between an application developer and a client, trust-based or contract-based governance mechanisms will be used, rather than authority-based governance mechanisms.

The transaction between developers sees different exchange conditions as with clients and developers. What is being exchanged is the development of an application. The process offers similar exchange conditions for the organizations involved in the transaction. Although there is often an owner of an idea, creating a power distance between the organizations. The developers are often part of a strong informal social system, or community, in which trust-based governance is preferred. This preference is reflected in their business relationship with other developers (Pittino & Mazzurana, 2013). Formal collaboration forms are rare and reputation seems important in the community. Trust between the developers prevents opportunistic behaviour and stimulates sharing knowledge and other resources (Jones et al., 1997). It can seem naïve that they don’t pay attention to contracts, but it is likely due to the lack of competition between developers for clients. The need to express the contingencies in a contract is low as the trust relationship is not likely to be impaired.

The collaboration can be seen as a form of vertical integration but with independent organizations involved in the collaboration. However, the dominant governance mechanism seems to be trust-based. A balance is sought for in which contract-based and authority-based governance mechanisms support the trust-based mechanism in the network dimension. This is line with previous research that has identified that multiple governance mechanisms can be used simultaneously (de Reuver, 2009; Geyskens et al., 2006; von Tunzelmann, 2003).

**H3:** In a relationship between an application developer and other application developers, trust-based governance mechanisms will be used, rather than contract-based or authority-based governance mechanisms.

The transaction between developer and patent holding company is different in the sense that the developer is not voluntarily entering into the transaction. The relationship is a pure market transaction as the patent holding company is referring to a patent infringement by the developer (Williamson, 1979). What is being exchanged is a licensing fee and a possible fine for respectively use and infringement of the patent.
The relationship is not to the benefit of the developer and should be prevented from occurring. Developers are often not aware that they are using a patented technology or just use it anyway, hoping that it will not be noticed. It can easily be called naïve or at least a sign of an immature market.

As the mobile ecosystem evolves, the developers that do pay attention to patent infringement are likely to survive. This will cause renewal in the ecosystem as organizations that lack focus on formal aspects of running an organization will become more successful than those that do not. An evolutionary mechanism of natural selection seems to mature the market and make formal obligations more important.

**H4:** In a relationship between an application developer and a patent holding company, a contract-based governance mechanism will be used, rather than a trust-based or authority-based governance mechanism.

From the perspective of the mobile ecosystem the organizations interact and have a mutual dependency on each other as they co-evolve towards a shared goal. The platform leaders need the clients to offer business opportunities to developers, and attractive applications to users. In a similar fashion, the developers need the platform leader to make a stable operating system that is bug free and offers unique possibilities for applications. This mutual dependency on each other can have an influence on the governance mechanisms that are more efficient and effective in the relationships.

A virtuous cycle is identified by the mutual dependency upon each other. It causes the ecosystem to evolve and thereby change the context of the mobile ecosystem. The market is becoming more mature, and thereby the hypotheses might change as formal control mechanisms become more important. Furthermore, the current governance mechanisms can limit the ability to produce radical innovations and thereby limit the growth potential of the ecosystem. The platform leaders can adopt a keystone strategy instead of a dominator strategy, thereby stimulating exploration instead of exploitation and stimulating the development of radical innovations. In such a relationship trust-based governance mechanisms will become more efficient and effective. Due to the speculative nature of these findings, the influence on hypotheses is not translated into new hypotheses.

### 7.1.3 Research question
The sub-questions and hypotheses work towards an answer for the research question. The research question of this study is:

**RQ:** How can a mobile application developer exert governance mechanisms in a mobile ecosystem to influence the relationship with other organizations in the ecosystem?

The exchange conditions determine if a governance mechanism is more or less efficient in a transaction. The conditions are influenced by the organizations in the transaction, and the product or service that is transacted. Influencing the relationship by the developer involves safeguarding, controlling, and coordinating the economic exchange. This can, for instance, be done to create an efficient and effective business relationship, or to maximize the value gained from the transaction.

Contract based governance mechanisms are an underdeveloped mechanism for most self-employed developers. They are specialists in programming but not entrepreneurs. They like technology but not doing administration, or legal issues, and therefore lack business skills. They are smart, enthusiastic,
and driven, but not always good in communicating with clients. As a result they face difficulties in commercializing their products, either as a service or as application. It can be surmised that the lack of business skills among many developers is a sign of an immature market. However, the app economy is growing rapidly and it is expected that a contract-based governance mechanism will see more practice in the future for self-employed developers. Nevertheless, it can be surmised that the smaller the organization, the less formal governance mechanisms are used.

Developers that have mastered these skills seem to have become more successful and often grow to be a developing company. With use of new contract forms such as Agile contracts, they give the client more control over the development as it is more transparent. Furthermore, it offers the developer more control over the process and input that is needed to work towards a result. Commercializing products and services is left over to project or account managers, and thereby the developers can do what they do best.

The governance mechanisms in the strong informal social system that developers often are member off reflects in the way they prefer to govern their relationships. The trust-based mechanism is a powerful governance mechanism as it allows the developer to handle with the uncertainty in the technology and the complexity of tasks. However, it might take time to develop inter-organizational trust. The trust relationship between the organizations places them on an equal level, as far as possible, which reduces the power distance and the ability to use a governance mechanism based on authority.

7.1.4 Contributions to theory
This research contributes to the understanding of governance mechanisms that can be used by developers in a business network from the perspective of a mobile ecosystem. Research from a perspective of a non-focal organization, often a SME, in an ecosystem is scarce. And this research adds to previous contributions (Holzer & Ondrus, 2011; Kokabha, 2012; Pittino & Mazzurana, 2013; Selander et al., 2013) as it offers insight into the governance mechanisms that are exerted in a relationships that a developer can have in a mobile ecosystem. This is relevant as in particular developers have limited resources and are thereby forced to acquire these resources elsewhere, often in their business network (Selander et al., 2013). If they can improve their ability to extract value from these relationships, they can have a competitive advantage over other organizations and are more fit for the evolving environment.

Second, this research contributes to the growing body of research on the app economy and the dynamics that are involved (Basole & Karla, 2011; Holzer & Ondrus, 2011; Jansen & Bloemendal, 2013). There seem to be many business opportunities that can be taken and the outlook of the market potential is enormous (ABI Research, 2013). Understanding how relationships are influenced by organizations within the business network is valuable for both practitioners and academics as it helps grasping co-evolutionary aspects that determine the faith of the ecosystem (Peltoniemi, 2005).

Third, the hypotheses that are identified offer scholars the possibility to start more elaborate descriptive research into the mechanisms and strategies that can be used to influence the relationship with other organizations. With knowledge of the influence they have, they can start predicting behaviour and develop strategies for their relationships. This can be used to steer the outcome of their relationship and may open the door to exert governance on a system level. This
research contributes in that sense as it supports previous findings for the need of a system level perspective on the relationships in a business network (Ahuja et al., 2011; Provan & Kenis, 2007). It is a first step in gaining insight into relationship governance by application developers in a mobile ecosystem.

Fourth, this research identifies both Apple and Google as platform leaders that adopt a dominator strategy. In contrast with research that suggest that Google can be seen as a keystone organization (Kokabha, 2012). In particular this contributes to research on platform strategies (Gawer & Cusumano, 2008) and business ecosystems roles and species (Levien & Iansiti, 2004). As the case study offers insight into the influence the strategies of platform leaders have on organizations in the mobile ecosystem. This can be used to refine current strategies and develop new ones. It also questions the subjective nature of these roles and asks for further empirical evidence on these roles.

Fifth, the high technological uncertainty limits the use of formal contracts in relationships between organizations involved with the development of applications (Vlaar et al., 2007; Williamson, 1979; Woolthuis et al., 2005). There seems to be a balance between creativity and control. Developers need to express creativity in their work to assure that an efficient and effective solution is found, however, this requires the client to give away much of the control on the project to the developer. Furthermore, the uncertainty creates difficulties for practitioners in describing all contingencies ex ante and as a result they often have small contracts that do not foresee important risks. This research has identified that in practice innovations in contracts have offered practitioners new methods in dealing with this uncertainty and the creativity versus control dichotomy. Agile contracts make the process more transparent giving more control to both developers as well as clients. This improves both the trust in the relationship between both organizations as well as the value that is extracted from the relationship.

Lastly, further empirical evidence is found for the non-mutually existence of governance mechanisms in a governance dimension (Bradach & Eccles, 1989; de Reuver, 2009; von Tunzelmann, 2003). In a governance dimension, a balance is sought for of the governance mechanisms that can be exerted. With knowledge of this balance, research can be performed in developing strategies for optimal balances in a collaboration. In a formal relationship with for instance a contract, trust-based governance mechanisms are used to assure that the organizations will live up to the obligations in the contract. Furthermore, an authority-based mechanism is used in the decision making process of the collaboration.

7.2 Limitations

This research tries to identify what governance mechanisms are exerted by mobile application developers to influence their relationship in a business network. With use of transaction cost economics and mobile ecosystem theory, the relationships are analyzed. It can be seen as a first step towards predicting behaviour and developing strategies for it. However, the chosen approach and theory inherently has certain limitations which are elaborated upon.

The process of application development is generalized as a project entailing only one step. However, previous research has already identified that within a project or process, different governance mechanisms are more or less effective depending on the phases and time within a project (de Reuver, 2009). Furthermore, the behaviour of the organizations in the relationship is kept constant,
assuming the organizations will not adopt any strategies in predicting behaviour. However, with for instance game theory, strategies can be developed to predict behaviour and adopt the governance mechanisms accordingly (Axelrod & Dion, 1988; Nalebuff & Brandenburger, 1997). Also platform theory can give more insight into the behaviour of platform leaders and the dynamics that are present within the platform (Gawer & Cusumano, 2008; Jansen & Cusumano, 2012). Not including these theories limits the view on the governance mechanisms and requires future research to gain a more complete view. The delivered hypotheses are a first step towards achieving this.

This research adopts both an ego-network approach as well as a whole network approach. This is possible with the use of a business ecosystem perspective. However, the limited influence that a developer can have in its business relationships, limits the influence in the network and thereby the use of the business ecosystem metaphor. Only developers that actively try to bridge structural holes seem to have influence throughout the network, apart from the relationship that are described in this research. The business ecosystem metaphor helps in explaining the co-evolving nature of the organizations that are involved in the platform. Furthermore, it explains the mutual dependency that the organizations have with each other. However, the application for the hypotheses is limited as the governance mechanisms are exerted in relationships between two organizations with only minor influence of system level implications. The analysis can be expanded by involving more practitioners in the research from different perspectives in the ecosystem. It will give a more complete view on the business network and the interdependencies that exist between the participants. The lack of interaction between platform leader and developers has created a strong community of developers who look to each other when trying to solve problems and issues instead of the platform leader. These tensions need to be researched on its own.

Organizations in the mobile ecosystem are mostly not aware that they are in an ecosystem, let alone of governance mechanisms. If they all would be aware of the ecosystem analogy and governance mechanisms, this can change the way they behave in the ecosystem. It is expected that this will cause the hypotheses to involve more formalized governance mechanisms, e.g. contract-based and authority-based.

As information with respect to business relationships is rarely codified, the use of a case study approach was limited to interviews and observations of community meetings. It was further supported by secondary data with respect to the mobile ecosystems of Apple and Google. And although the interviews were long, the amount of different interviews is low (8). This limits the internal validity and reliability of the method as the actual process of governance in a business relationship is only partly observed and based on the interviews with primarily developers. The validity of the results can be improved by testing it with industry experts or scholars, or with practitioners in a workshop. Furthermore, a survey can be conducted among a wider audience to further validate the findings in support of the hypotheses.

The single-case study method lacks validity as it is only tested in one domain. The domain is in particular interesting as it is a growing and still immature market. It is becoming of interest of scholars (Angeren et al., 2013; Eaton et al., 2011; Ghazawneh & Henfridsson, 2013; Selander et al., 2013) but the generalizability of the findings is limited to this domain. Selecting these types of species and the perspective of a developer inherently limits the model to ecosystems with similar types of relationships. The model can applied in for instance a domain of software development such
as the ecosystem of PlayStation, or Microsoft. These domains have similar relationships in which the market is already mature. It will yield in a better validation of the model and will allow for the development of more generalizable hypotheses. More general statements with respect to roles and species in a business ecosystem will require extensive research in which the different perspectives of organizations in the business network are explored.

The interviewees have been developers and a client of developers. Whereas the relationships that have been researched involved more parties, e.g. platform leaders and patent holding companies. Within a relationship, two organizations determine how the relationship evolves and they can have a different perspective on the situation. Not involving the other perspective can give a biased view on reality. To support the findings, interviews can be conducted with managers from the platform leaders or patent holding companies. For instance managers that are involved with the strategy of the platform or market place will yield a more complete view on how the business relationships are formed and governed. Furthermore, more interviews with clients will yield in a stronger support of the hypotheses and improve the generalizability of the findings.

The developers are often in the size range of SMEs. This limits them in their availability to resources but offers them several advantages in for instance speed of decision making and high levels of interpersonal trust. However, in this research only in the relationship with platform leaders and patent holding companies have the differences in characteristics created differences in the relationship between the organizations. Between other developers and clients, the characteristics are similar and the extent to which it influences the developed hypotheses is therefore limited.

The theory to analyze the relationships in this research has been transaction cost economics. This assumes that the organizations involved in the relationship are bounded rationally and at least some are opportunistic. However, when looking at how developers exert governance in their relationships, it can be argued that they are not always opportunistic and sometimes rather are idealistic.

This research assumes that governance mechanisms arise by the influence of exchange condition, and to a minor extent is influenced by business ecosystem aspects on a system level. Using exchange conditions is in line with research by Bradach & Eccles (1989), but there can be many more different predictors of governance mechanisms. Such as for instance the innovation phase (de Reuver, 2009) or structural properties of the network (Uzzi, 1997). When performing future research on the hypotheses, involving other factors than exchange conditions and system level aspects, will give a richer and complete view on the relationships and the influence that developers can have in it.

Lastly, this research only identified the governance mechanisms that are exerted to influence a business relationship. It does not provide in a measure of the influence and how it can have an impact on the organizations involved. This will require research in which the performance of the organization is measured with longitudinal data. Furthermore metrics should be operationalized to the extent of the governance mechanisms that are used.

7.3 Recommendations
The hypotheses are the starting ground for future research. The results have both implications for practitioners and academia, and are detailed upon in this section.
7.3.1 For practitioners

The research has identified the governance mechanisms that are exerted by mobile application developers in a mobile ecosystem to influence their relationship with other species in their ecosystem. With knowledge of these mechanisms, a practitioner can assess the relational risks of the transaction with another organization. If large differences exist in how the organizations in the exchange prefer to exert governance mechanisms, this can indicate if problems can be expected in the relationship. For instance, a client that prefers working with contracts and is suddenly faced with a developer that lacks the skills to draft a contract, can be incentivized to show opportunistic behaviour. The developer must compensate the lack of contract-based governance mechanisms with a strong trust bond to safeguard and control the exchange.

In general the app economy is relatively immature and can see many improvements in how transactions are governed. Although contracts are difficult to develop ex ante, they reduce the chance of opportunistic behaviour by the other organization and help control possible debtors. Innovations in legal contracts, such as Agile contracts, can make this possible. In particular the self-employed developer is advised to gain knowledge on this field or find a partner that has a similar mind-set and is willing to help build up a company with more skills then mere application development. However, the practitioners should be aware that they are constrained in their ability to actually change governance mechanisms, as they might lack leverage to do so unilaterally (de Reuver, 2009, p.115). In general, developers are advised to formalize their relationships to safeguard the transaction and limit the incentives for opportunistic behaviour.

The developers that act as bridge players in the ecosystem are in the position to appropriate on the economic exchanges that take place in the business network. This can be a valuable side-path to developing as there seems to be a gap between developers and clients. They can strengthen their position by offering (offline) events for clients and possible clients and thereby connecting developers and clients with each other. These events can be used in a similar fashion as now is common among developers, and thereby stimulate sharing knowledge but with involved of clients. In a similar fashion should developers be stimulated to participate in these events. It opens the door to new business opportunities and can help them in obtaining solutions to problems that they might face.

The patent holding companies in the ecosystem are continuously on the lookout for any infringements of their patents. By improving control of the development process and performing compliance management, can a possible patent infringement be prevented. This will prevent the developer from unnecessary losses. As it is difficult for a single developer to perform these activities, there could be a benefit of a group of developers or a community that actively monitors frequently used technological functionalities that have been patented. Thereby providing in a database for those that are involved with development. Larger organizations seem to pay more attention to patent infringement as they are aware that any licensing fee can seriously damage profits that have been generated.

From the perspective of developers the platform leaders are seen as dominators. There is little to no influence by the developer in the relationship with the platform leaders. However, the application developers are of grave importance to the platform leaders as the applications are one of the key assets that make a platform interesting to users. It can be argued that platform leaders act arrogant
as they standardize the relationship they have with developers and do not give any support on the platform. Platform leaders are recommended to foster the way developers are increasing the value of their platform, and should stimulate the value creation and development by offering more support to their platform.

7.3.2 For academia
The research has been performed in the domain of the mobile industry, as a single-case study. More elaborate research from a different perspective in this ecosystem can improve the validity of the hypotheses. For instance, by conducting interviews with platform leaders, clients, and patent holding companies. Furthermore, academics can improve the generalizability of the findings by conducting a similar research but in a different domain, such as the ecosystem of the PlayStation console or the software ecosystem of SAP, Oracle.

Furthermore, the difficulty that developers experience in describing the process ex-ante seems to relate to the lack of legal contracts that are used in practice. Research on how developers can use their strong trust relationship while incorporating contracts in their business can prove a valuable insight. However, this relation needs further support and can be subject to future research.

In line with research by Ritter (2007), the balance of governance mechanisms used in a dimension can be visualized on three axes in a triangle. By using this method, profiles can be developed, with conditions that determine when a certain balance is likely to emerge. This can be used by an organization as assessment in advance of the possible relationship, to determine possible risks in the relationship and develop mitigating measures. Although the method has been proposed in literature, it has not seen many application and can prove to offer valuable insights to scholars as well as practitioners.

The hypotheses that have been proposed are subject to future research. With descriptive or quantitative research a next step in understanding the governance mechanisms in relationship with species in a mobile ecosystem can be made. Once the mechanisms find enough empirical evidence, the research can be expanded with other theories such as game theory to gain insight into the behaviour of organizations surrounding the transaction. It can be used to develop scenarios and strategies for the participation of SMEs in an ecosystem. It will require involving more predictors of governance mechanism in a relationship next to exchange conditions.

This can be achieved by performing a meta-analysis with respect to predictors of governance mechanisms, and assessing the insight that these predictors give. If relevant they must be involved in a larger research in which the dynamics of governance mechanisms are explored (de Reuver, 2009) allowing the hypotheses to be refined and the validity to be improved. Once the hypotheses have enough support, a survey among practitioners and academic experts that are involved in the mobile industry can be used to measure the extent that the mechanisms are used. An online questionnaire can be send to a sample consisting of organizations categorized in the different types of species. The sample can be spread worldwide, although it might be difficult to find organizations that are willing to participate far outside the Netherlands. The participants must engage in inter-organizational relations, for any results on governance to be meaningful.

Measures for governance mechanism have been proposed and are used by scholars in previous research (de Reuver, 2009). The metrics for predictors of governance mechanisms depend on the
predictors that are added to future research. Exchange conditions have been operationalized in earlier work by Jones et. al (1997). The species of the organizations in the ecosystem can be identified with help of industry experts and can be cross validated in a survey. The hypotheses can be tested in mobile ecosystems but also in a domain that has similar organizations in the business network. For instance in software development, such as game consoles of Playstation and Xbox, but also on Microsoft’s Windows 8 platform. Once operationalized and measured, the results can statistically be analysed.

7.4 Reflection
In hindsight different steps might have been taken in the process of this research. A reflection on the approach and the research finalizes this thesis.

7.4.1 Reflection on approach
The theory used in the research is good in analyzing the relationships and to identify what is happening. Equally interesting, is it to develop strategies to predict the behaviour surrounding a relationship. This, however, is only possible when knowledge of the mechanisms is present. The dearth of research with a viewpoint of a non-focal organization within an ecosystem made this impossible and therefor this approach was chosen.

The study experienced practical issues in performing the actual case study. A case study based upon information from the internet yielded in results that were subject to bias and incomplete. The information with respect to the business relationships of organizations is often not codified and if it is, it is not that easily accessible. As this research is performed from the TU Delft, organizations are more reluctant to share this information. The result is that only interviews and observations are the source of information, limiting the in-depth analysis and applicability of the case study approach. Reflecting the chosen plan of approach, the interviews could have started earlier to allow more interviews to be conducted during the time-span of the thesis.

Also, using a single-case study limits the validity of the results. It would be preferred to adopt a multiple-case study method but was found unfeasible given the time constraints to this research. And by exploring the world of application developers in the Netherlands, in particular within the vicinity of Amsterdam, the domain was isolated. However, it was limited to the extent that developers were willing to show me around in their world. Some developers were from abroad, ranging from Silicon Valley, to Brazil, making validation possible from the findings in this area. But although they were really helpful, it only gave insight into their informal social network. Grasping an objective view on the world of application developers required more interviews or a survey to be conducted in different communities. This however, was beyond the scope of this project given the time constraints.

In hindsight, also the data analysis method could be improved as it did not involve proper coding with use of qualitative research software. Instead, pattern matching is adopted that could be subjective to bias from the researcher and lacks internal validity.

7.4.2 Reflection on process
When starting the thesis, the objective and scope of the research were too vague and broad. The perception was that the project should deliver groundbreaking results and both theory and empirical domain should be turned inside out. However, this led to a research proposal that lacked focus and
rigor which required much ‘discovering’ along the way. This can be called naïve as in the feedback of all reviewers I was warned for this. In particular, not selecting to enter a domain until halfway the process, after the theoretical framework was developed, caused difficulties once the first results from interviews came in. During the development of the research proposal I heard of the possibility to conduct research within the domain of Philips Healthcare. This seemed interesting and this was the domain that was explored in the beginning of the project. However, once the lead to this domain was approached to actually enter this domain three months later, this was no longer possible as other researchers had already commenced in this domain.

In a short period of time a new domain was chosen, with grateful help of the lead, to prevent any further delay. However, the initial proposition that depending on the keystone or dominator specie, different governance mechanisms should be used did not hold as there was no keystone organization in the domain. Furthermore, the developer had a relationship with the platform leader that involved a large power distance. It was not possible to influence the relationship it had and, hence, the research objective would yield no ‘interesting’ results. In stark contrast to the earlier perception of the goal of the research project. Also, quite soon in the interviews, did it come apparent that the developers were not aware of mobile ecosystems, let alone governance. This limited findings to the extent to what could be operationalized.

But although the lack of focus and rigor can be blamed for this, there is also a large gap between empirical evidence and theory on business ecosystems and platform theory. Having read so many articles on these topics in the first few months, the propositions were developed purely on a theoretical foundation. Pre-testing the propositions in an early stage would have immediately given the possibility to change either domain or conceptual framework as it turned out that the propositions would not hold. It required altering the conceptual framework, changing the unit of observation, and adapting the interview protocol to gain different results.

Another consequence of the long period involved with the development of the literature analysis was the demotivation it caused. A continuous process of judging if the theory would be good enough or yield interesting results led to an unproductive period that caused a delay in the overall process. Combined with working activities that are more easy and at that time more fun to do, it was easy to get distracted and do something else instead of working on a thesis. By forcing to make decisions with respect to theory and domain, was I able to step out of this spiral and actually start building the thesis in a constructive way.

Nevertheless, the results of this thesis are only a small step into discovering how an application developer can exert governance mechanisms and what influence they have in a relationship. With for instance platform theory, two-sided markets, a resource based view, and game theory, will a more elaborate image on the actual concept of governance be created. It will allow strategies and scenario’s to be developed to predict outcomes of transactions and steer the evolving business network of organization from a micro economical approach.

In retrospect, I would not have started the actual thesis until both the theory and the domain were decided upon and approved by all involved. Furthermore, I would have started with a more clear problem, and problem owner, to guide the overall process. Also If I had more time I would have elaborated the in-depth analysis and use a computational data analysis method. This will give much more rigor to the validity of the research and give a stronger foundation to the hypotheses that are
proposed. Furthermore, the hypotheses could have been tested as there are already measures of exchange conditions and governance mechanisms. The interviewees seemed willing to share a possible survey among their community. Performing these activities would yield a research that in my opinion will give a more complete view on the domain that is under study and would be a more solid contribution to theory and future research.
References


Appendix A  Interview protocol

The interviews are performed with the use of an interview protocol. This appendix describes the protocol that is used for these interviews. The following paragraph gives an email and accompanying short description and goal that is send to the interviewees beforehand the interview. The next paragraph gives the protocol and questions. Depending on the experience and knowledge of the interviewee, the questions have been adapted to suit the situation.

Accompanying email

I would like to ask your cooperation for an interview. Small- and medium sized enterprises (SMEs) often cooperate with large central organizations. In my master thesis project at the Technical University of Delft, I investigate how SMEs can shape and influence such relations.

One of my supervisors, Mark de Reuver, informed me that you might be able to help me gain insight into the world of app-developers and how they work together with larger organizations. I hope you will be willing to talk with me for about an hour.

<Or>

I was informed by <person> that you might be able to help me gain insight into the world of app-developers and how they work with larger organizations. I hope you will be willing to talk with me for about an hour.

In the attached file I have written a short description of the project I am working on and the goal I have for the interview.

I hope to hear from you soon. If you have any questions please don't hesitate to ask them.

Attached file:

Description of project

In modern society organizations are increasingly becoming dependent upon each other. For instance as access to resources such as information or money, or for the development of a product. As a result, small- and medium sized enterprises (SMEs) have to, or wish to, cooperate with large central organizations. In this project, performed from the Technical University of Delft, it is researched how SMEs can try to influence their relation with a large, central, organization.

Within industries, platforms with a few central, large, organizations are becoming a common phenomenon. For instance Apple or Google around the iPhone or Android smartphones. These organizations are central in the network around the platform because they have an important and large contribution to the network. Due to their size, or contribution to the network, they are able to choose with whom they wish to cooperate.

Various reasons could be the cause for SMEs to wish to cooperate with such central organizations. The SMEs however, have certain characteristics that limit them in their actions, such as restricted (financial) resources. As mentioned, this research tries to gain
insight into the options available to an SME to influence their relation with a central organization.

**Goal interview**

The interview I wish to take serves two purposes. Firstly I would like to know how often SMEs cooperate with these large, central, organizations and who these organizations are. Furthermore, I would like to know what mechanisms are used in the relation. Is everything, for instance, detailed in a contract or is it based upon good trust?

Secondly, I would like to know how the network of companies within the App-developer scene is organized. With this I hope to gain insight in possible candidates for new interviews.

**Protocol**

The procedure has been to make notes during the interview, a report of the interview is written immediately, or as soon as possible, after the interview. The report has been send to the interviewee as result and for approval of the findings.

The questions have been divided in 5 chapters, each with a main question to be answered by the sub-questions in that chapter. The chapters and main questions are:

1. **Broad exploration of topic**
   - Can you identify species in your business network?

2. **Relationships with organizations**
   - Do you have a relation with organizations in the mobile ecosystem (or done business with)? Or know companies that do?

3. **Exchange conditions**
   - Can you explain the conditions surrounding the relation?

4. **Governance mechanisms**
   - What mechanisms of governance do you use in your relation with other organizations? And do you feel you can influence and shape the relationship?

5. **Finalizing interview**
   - Is there anything I should know relating to my topic? And would you know other organizations I can speak to?
Questions

1 Broad exploration of topic

Main question: Can you identify species in your business network?

- Have you heard about business ecosystems?
  - Short introduction on perspective on business networks.
- Can you mention a company in the business ecosystem that is the most important and prominent in the network?
- What makes this company so important and prominent?
- Are there other companies alike?
- What makes them so important and prominent?
  - Short explanation of roles in the ecosystem, explaining that my subjects are companies that do business (have a relation) with dominators and keystones.
- Do you know what governance is? And in a business ecosystem?
  - Short explanation of governance in a business ecosystem.

2 Relations with species in the ecosystem

Main question: Do you have a relation with organizations in the mobile ecosystem (or done business with)? Or know companies that do?

- Have you done business with other companies in the ecosystem?
- If no, would you like to do business with them? Why?
- How would you describe their role in the ecosystem?
- Do you know other companies that frequently do business with these companies?

3 Exchange conditions of relation

Main question: Can you explain the conditions surrounding the relation?

- How many times have you done business with a specie in the ecosystem? (frequency of transaction)
- Can you explain the incentive for the organization to work with you? (asset specificity)
- Would you consider yourself a specialist with capabilities that the keystone or dominator does not have? (asset specificity)
- Are the steps towards the end result clear beforehand? (environmental uncertainty, complexity of tasks)
- Can you give a measure of the uncertainty surrounding the technology? (technological uncertainty)
- Upon delivery of the end product is the client most often satisfied or do you have discussions about the end product sometimes? (behavioural uncertainty)

4 Governance mechanisms

Main question: What mechanisms of governance do you use in your relation with other organizations? And do you feel you can influence and shape the relationship?
• When you agree to cooperate with the client (dominator/keystone), do you draft up a legal contract? (*market mechanism (MM)*)
• How extensive is this contract? Would you specify as much as possible beforehand? (*MM*)
• Can you influence what is in the contract? And to what extent do you feel you can influence this? (*MM*)
• Does the client require the lowest price? Why? (*MM*)
• Is there trust between you and the client? Does that influence your cooperation? (*network mechanism (NM)*)
• Do you have clients that work with you based upon trust to a large extent? (*NM*)
• Is the reputation of the client important to you? Would (or do) you use it as referral? (*NM*)
• Would you prefer working based upon trust that you will deliver an end product or do you like to draft everything up in contract? Can you explain a possible balance between the two? (*NM*)
• Does the client think the same way? (*NM*)
• How dependent is your organization on the client? Has this changed over time? (*hierarchy mechanism (HM)*)
• Is there an equal relationship with the client? Who makes decisions? (*HM*)
• Are there rules surrounding the relation? How strict are they? (*HM*)
• Does the way of collaborating work well? Would you like to change anything? (*Performance*)
• Do you find the collaboration successful? (*Performance*)

5  Finalizing interview

Main question: Is there anything I should know relating to my topic? And would you know other organizations I can speak to?

• How did you experience this conversation?
• Did I omit any questions I should have asked?
• Do you have any questions yourself?
• Are there any other organizations I should speak to?
Appendix B  List of interviewees and observations

<table>
<thead>
<tr>
<th>#</th>
<th>Interviewee</th>
<th>Company</th>
<th>Specie</th>
<th>Function</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Peter Robinnet</td>
<td>Bubblefoundry</td>
<td>Developer</td>
<td>Founder</td>
<td>29 May</td>
</tr>
<tr>
<td>2</td>
<td>Pieter Lekkerkerk</td>
<td>EscolherSeguro</td>
<td>Developing company</td>
<td>Co-founder / CEO</td>
<td>29 May</td>
</tr>
<tr>
<td>3</td>
<td>Robert Atkins</td>
<td>E-Buddy</td>
<td>Developing company</td>
<td>Developer</td>
<td>29 May</td>
</tr>
<tr>
<td>4</td>
<td>Koen Mostert</td>
<td>Concapptual</td>
<td>Developer</td>
<td>Co-founder</td>
<td>9 June</td>
</tr>
<tr>
<td>5</td>
<td>Jurren Pen</td>
<td>Concapptual</td>
<td>Developer</td>
<td>Co-founder</td>
<td>9 June</td>
</tr>
<tr>
<td>6</td>
<td>Anonymous</td>
<td>Xebia</td>
<td>Developing company</td>
<td>Project Manager</td>
<td>19 June</td>
</tr>
<tr>
<td>7</td>
<td>Anonymous</td>
<td>Xebia</td>
<td>Developing company / developer</td>
<td>Developer</td>
<td>19 June</td>
</tr>
<tr>
<td>8</td>
<td>Marcel Groenewegen</td>
<td>Supplier-DNA</td>
<td>Client</td>
<td>CEO</td>
<td>2 July</td>
</tr>
</tbody>
</table>

Table B1; Overview of performed interviews.

<table>
<thead>
<tr>
<th>#</th>
<th>Event</th>
<th>Location</th>
<th>Species</th>
<th>Function</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Appsterdam</td>
<td>Café Bax Amsterdam</td>
<td>Developers / developing company</td>
<td>Social meeting</td>
<td>29 May</td>
</tr>
<tr>
<td>2</td>
<td>Cocoaheads</td>
<td>Xebia</td>
<td>Developers / developing company</td>
<td>Knowledge sharing</td>
<td>19 June</td>
</tr>
</tbody>
</table>

Table B2; Overview of observations at developer events.