Exploring Platform Ecosystems: A Comparison of Complementor Networks and their Characteristics

Joey van Angeren
J.vanAngeren@uu.nl

Utrecht University
Department of Information and Computing Sciences
Master in Business Informatics

MSc. thesis submitted under supervision of:
First supervisor: Dr. Slinger Jansen (Utrecht University)
Second supervisor: Prof. dr. Sjaak Brinkkemper (Utrecht University)

November 2013

Abstract

Owners of software platforms are increasingly dependent on developers of complementarities. As the proprietary platform itself exhibits elementary or generic functionality, platform owners depend on a complementor ecosystem populated by third-parties. As such, the ecosystem became a pivotal determinant for the success or failure of a software platform in platform-based competition. At present, little is known about mechanisms at play in proprietary platform ecosystems, and it remains unclear how these ecosystems differ from each other across firms and platform types. Addressing this deficiency, this thesis investigates and contrasts four proprietary platform ecosystems through statistical and network analysis.

The research compares the ecosystems that exist around Google Apps, Google Chrome, Office365 and Internet Explorer, with data obtained by means of automated app store data extraction and interfirm relationships obtained from company websites and CrunchBase. Results show similarities among the four proprietary platform ecosystems. The ecosystems are sparsely connected and highly centralized, since 3.18% to 29.82% of complementors initiated interfirm relationships. Furthermore, the ecosystems are predominantly populated by complementors that limitedly commit to application development, the average number of applications developed per complementor ranges from 1.34 to 2.18. This is especially apparent in the Google Apps and Office365 ecosystems, that display strong characteristics of power law scaling in the distribution of the number of applications developed per complementor.

Despite the apparent similarities, Google platform ecosystems harbor a greater population, Microsoft ecosystems are more densely interwoven by interfirm relationships and web browser ecosystems display sheer connectivity. The differences are attributed to the lower entry barriers to complementary markets imposed by Google, the positive relationships between the number of applications a complementor develops and the number of interfirm relationships it initiates, the active partner enablement by Microsoft and the degree of homogeneity of the species in the ecosystem. The research method and results presented in this thesis can be used by practitioners as a reference to evaluate their structural position in the ecosystem, whereas it provides researchers with a quantification of ecosystem characteristics and a step towards better understanding of forces at play in proprietary platform ecosystems.

Acknowledgements

This Master's thesis is all about the fascinating topic of proprietary industry platforms and their surrounding ecosystems populated by developers of complementarities. It visualizes and contrasts four proprietary platform ecosystems, and is among a limited number of studies in existing scientific literature to study webs of interfirm relationships in a usually 'closed' setting. As such, the research presented in this thesis unites the perspectives of software platforms, ecosystems, management and strategy.

The research presented in this Master's thesis is the result of an extensive project that lasted for fourteen months. Throughout this period, I have been accompanied and supported by numerous people. To start with, I would like to thank my first supervisor Slinger Jansen, who I have intensively worked with for multiple years within and beyond the scope of the research described here. Slinger, thank you for your contribution in acquainting me with the scientific community, our intensive and thought-provoking feedback sessions, your contagious enthusiasm and your patience in the time-consuming initiation of this project. Next, I would like to thank my second supervisor Sjaak Brinkkemper. Sjaak, thank you for the feedback, reviews and advice. I am grateful for your guidance in formalisms and definitions, which led to formal definitions of principal concepts in this thesis.

I would also like to thank Peter Buxmann and Thomas Widjaja from Darmstadt University of Technology for providing me with the basic infrastructure of their web crawler. This infrastructure formed the basis for the web crawler that was used in the automated app store data extraction process, and as such played a principal role in the execution of this research project. Moreover, I would like to thank my friend, colleague and business partner Vincent Blijleven. Our intensive collaboration has led to multiple scientific publications, strong professional development and ultimately the founding of our own consultancy start-up Think Ecosystems. Vincent, thank you for the continuous feedback, support and intensive collaboration in the last couple of years.

Of course, my gratitude also goes to my friends, family and girlfriend. They have surrounded me for many years, supported and encouraged me whenever necessary. I am very grateful for all of that.

Joey van Angeren
November 2013

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The software industry has become increasingly interconnected and interdependent. Not only do software vendors cooperate with an array of other organizations in the software industry to deliver their products (Jansen, Brinkkemper, & Finkelstein, 2007), the value of this software product is also determined by the network of organizations that exists around it (Gawer & Cusumano, 2002). The value of the Android mobile operating system, for example, increases when more software companies and individual developers build applications for this platform. Similarly, the value of the SAP enterprise resource planning platform increases as SAP succeeds to attract more partners that build domain-specific extensions for this platform. The dependency between the value of a platform and the network of actors that forms itself around it implies that all actors within a network share the fate of the network as a whole (Den Hartigh, Tol, & Visscher, 2006), even if not all actors carry equal responsibility for the establishment, management and orchestration of the network.

Under pressure of platform strategies and resulting third party platform extensions, networks of interrelated organizations form themselves around products, platforms, technologies or software organizations, which are referred to as software ecosystems (Messerschmitt & Szyperski, 2005). Jansen, Brinkkemper, and Finkelstein (2009) define a software ecosystem as "a set of actors functioning as a unit and interacting with a shared market for software and services, together with the relationships among them" (p. 2). Herein, actors are the entities (e.g. companies, standard bodies, individuals) that populate the ecosystem because they participate in, or contribute to it (Van Angeren, Blijleven, Jansen, & Brinkkemper, 2013) Within a software ecosystem, one or more central actors share the responsibility for maintaining or expanding the network, called keystones (Iansiti & Levien, 2004a). At least one of the keystones is the owner of the product, platform, market or technology around which the ecosystem exists (Jansen, Brinkkemper, & Finkelstein, 2009). Furthermore, an ecosystem can be contained in another ecosystem – a firm ecosystem may contain multiple platform and technology ecosystems. The Google software ecosystem, for example, among others consists of the Android ecosystem, the Google Maps ecosystem and the Google Apps ecosystem.

Software platforms are typical in that they are regarded as a foundation technology that brings together multiple parties (Gawer & Cusumano, 2002). According to Baldwin and Woodard (2009), a platform consists of three core elements:

- **Foundation technology:** Business application suites, (mobile) operating systems, web browsers
- **Interfaces:** APIs, component buses, service buses, glue components
- **Complements:** Apps, plug-ins, add-ons, extensions

A platform ecosystem consists of all parties that build complements to the software platform. Owners of software platforms have mechanisms and tools at their disposal to attract complementors, to create extensions for a platform and thus expand the software ecosystem around it. Among others, platform owners can create partnership models for either the platform or the organization as a whole (Van Angeren, Kabbedijk, Jansen, & Popp, 2011; Van Angeren, Kabbedijk, Popp, & Jansen, 2013), market their platform (Jansen & Cusumano, 2012), create APIs or other platform extension mechanisms (Alspaugh, Asuncion, & Scacchi, 2009; Baldwin & Woodard, 2009; Jansen, Brinkkemper, Hunink, & Demir, 2008) and offer extension markets such as application stores and plug-in listings (Jansen & Cusumano, 2012). Apart from attracting complementors, these mechanisms also aid in interrelating different actors in the ecosystem. Actors, for example, establish technological partnerships or alliances with other actors, or a complementor can depend on product or service functionality offered by another complementor (Baldwin & Woodard, 2009). Subsequently, a densely interconnected platform ecosystem is established.

At present, the product software industry is characterized by platform-based competition between different organizations, such as the grasp for market share among providers of mobile operating systems. According to Cusumano (2012), there are two interconnected determinants that influence this competition: (1) platform strategy and (2) the prosperity of the ecosystem around the platform. The prosperity of an ecosystem is determined by its capability to create opportunities for its complementors, and their capability to deliver meaningful artifacts, while maintaining the capability of surviving disruptions (Iansiti & Levien, 2004b).

1.1 Problem Statement

Over the last decade, the attention for the adoption of ecosystem theories in the software industry has been vastly increasing (Barbosa & Alves, 2011; Hanssen & Dyba, 2012). While researchers elaborate on the importance of direct and indirect network effects and platform-based competition in the software industry, so far little attention is directed at in-depth studies of ecosystems to provide for characterization and illustration of network effects. Accordingly, at present there is no insight in the factors that shape platform ecosystems, their network structure and interdependence among actors, apart from the findings presented by Iyer, Lee, and Venkatraman (2006) stating that the structure of the entire software sector remained stable between 1990 and 2001. While it is known that a firm ecosystem can contain multiple platform ecosystems, so far the relationship between the characteristics of these respective ecosystems remains elusive. Software companies, such as Apple are perceived to be closed and inaccessible because of their numerous proprietary standards and retained governance structure, compared to their more open siblings (Boudreau, 2010). Microsoft in turn is known for its active partner management, whereas Google only recently started operating partnership models. The perceived openness or partner management in turn could reflect in the characteristics of the software ecosystem. Gawer and Cusumano (2002) and Cusumano (2012), however, argue there is a relationship between platform strategy – and thus the characteristics of the software platform – and the ecosystem that exists around the platform. Accordingly, it is apparent that the Android platform differs from the Google Apps platform, but what the exact influence of this difference is on the respective platform ecosystems remains unclear. Moreover, it remains unclear if the ecosystems share equal characteristics or that comparable platform ecosystems (e.g. Google Apps and Microsoft Office365) share similar characteristics. In sum, these discrepancies can be captured in the following problem statement:

It is unclear what factors shape the network structure of platform ecosystems. Accordingly, it remains unclear how platform owners can manipulate the network structure in their complementor ecosystem, which among others hampers innovation, niche creation and collaboration. Without insight into the factors that shape platform ecosystems, the effect of ecosystem orchestration mechanisms remains unknown.

To address the current discrepancy, this research project compares the network structure of four platform ecosystems by means of a theory building multiple case study (Eisenhardt, 1989). In order to do so, all platform ecosystems are represented as a constitution of complementors and the business relationships among them. To benefit comparability two case companies have been selected, for each of which two platform ecosystems are subject of study. Requirement is that the sets of platforms from both companies must be comparable. If, for example, a mobile operating system and a CRM platform are subject of study for the first case company, the second case company requires a similar platform portfolio. To compare the network structure of the four respective platform ecosystems, network metrics are adopted from the research domain of network analysis (Burt & Minor, 1983; Scott, 2000; Hanneman & Riddle, 2005). As illustrated in previous studies by Iyer et al. (2006), Basole (2009) and Kabbedijk and Jansen (2011), network analysis can be applied to software ecosystems as it provides metrics that quantify characteristics of ecosystems. Among others, this analysis will aid in the identification of influential subgroups and clusters in the ecosystem, building on the classification of actors that are found in ecosystems, and furthermore provides a step towards the determination of the current and future position of the keystone within the ecosystem. Accordingly, this research aims to build on the practices of ecosystem orchestration (Jansen, Brinkkemper, & Finkelstein, 2009) and platform leadership (Gawer & Cusumano, 2002) alongside the characterization of software ecosystems, to the benefit of both academia and practitioners.

1.2 Research Questions

The main research question answered in this thesis is: *"How do actors organize themselves in a proprietary platform ecosystem, and what factors influence the hereof resulting network structure?"*. To facilitate in providing an answer to the research question, the following sub research questions have been formulated:

1. **How can network analysis be applied to analyze proprietary platform ecosystems?** – Software ecosystems are constituted of actors and relationships among actors, and can thus be visualized as a network. The current position of an ecosystem orchestrator, for example, can be determined based its centrality within the ecosystem (Jansen, Finkelstein, & Brinkkemper, 2009). Contrary to research domains such as knowledge or social networks – where network analysis is applied on a regular basis – few previous studies have addressed network analysis as an instrument of strategic ecosystem analysis. To make ecosystem characteristics quantifiable and measureable, this research question is answered by the definition and operationalization of network metrics (Burt & Minor, 1983; Scott, 2000; Hanneman & Riddle, 2005).

2. **What type of actors can be distinguished in a proprietary platform ecosystem based on their role within the network?** – Regarded from a high

abstraction level, an ecosystem consists of three types of actors: (1) keystones that provide the foundation for (part of) the ecosystem, (2) dominators that assimilate (e.g. through mergers and acquisitions) and eliminate other actors in the ecosystem, and (3) niche plays being the companies that provide complements for the foundation of the ecosystem (Iansiti & Levien, 2004b; Van Angeren, Blijleven, et al., 2013). On a lower abstraction level, however, different types of actors can be distinguished based on their role within the network. A broker, for example, is an actor that connects two sets of actors within an ecosystem (Iyer et al., 2006). A classification of such actors provides an answer to this research question.

3. **How do the characteristics of the firm ecosystem influence the network structure of its containing platform ecosystems?** – The total firm ecosystem is constituted of different platform ecosystems, that form around the platforms within the platform portfolio of a software company. Whereas one company is perceived to be relatively open and accessible, other companies are perceived to be closed due to their proprietary standards, ecosystem border definition and rigid ecosystem governance. The degree of openness has an influence on the network structure of the firm ecosystem. The influence of the degree of openness on the containing platform ecosystems is examined by comparing the network structure of the two selected platform ecosystems across case companies.

4. **What is the influence of the type of platform on the network structure of the platform ecosystem that exists around it?** – Researchers such as Gawer and Cusumano (2002) and (Baldwin & Woodard, 2009) acknowledge similarities between different software platforms. However, none of the authors examined the way in which the similarities reflect in the characteristics of different platform ecosystems, such as mobile operating systems. Accordingly, this research question is answered by means of a comparison of complementor ecosystems that form around a similar platform.

To provide an answer to the (sub) research questions, four platform ecosystems will be investigated by means of a document study, data extraction from app stores and manual reconstruction of interfirm relationships. A web crawler – that combines data scraping scripts in Java programming code – aids in identifying all actors that currently offer one or more complements in the extension market of a platform. Afterwards, relationships between the actors will be identified manually by visiting the company website of each individual actor. The aim herein is to identify partner, technology and alliance relationships. Additionally, business relationship databases such as CrunchBase are queried to identify additional relationships. The identified data will be used to draw network graphs of each individual ecosystem, which will then be subject to network analysis, and cross-case comparison.

1.3 Relevance

1.3.1 Scientific Relevance

The research domain of software ecosystems is relatively new and therefore still leaves research opportunities unaddressed (Barbosa & Alves, 2011; Manikas & Hansen, 2012). Previous research has been predominantly directed at uncovering mechanisms in open ecosystems because of their relative accessibility and transparency. While systematic literature reviews by Barbosa and Alves (2011) and Manikas and Hansen (2012) revealed that the attention for software ecosystems is increasing, still limited attempts have been made to describe and visualize entire ecosystems or subsystems of ecosystems. Exceptions to this in open ecosystems are research projects conducted by Kabbedijk and Jansen (2011) and Mens and Goeminne (2011), respectively on the Ruby and Gnome open source communities. By visualizing the Ruby ecosystem, the authors found that the ecosystem was coordinated by multiple influential actors, rather than just Ruby as an organization. These findings have implications for the management of the Ruby platform and community. With regard to closed software ecosystems, Basole (2009) visualized the mobile ecosystem and Iyer et al. (2006) presented the results of a longitudinal study, revealing that the overall network structure in the software industry remained stable between 1990 and 2011. More recently, Iyer (2012) described a systematic method for ecosystem visualization, that was illustrated by drawing network graphs of the mobile payment ecosystem. Burkard, Draisbach, Widjaja, and Buxmann (2011) then, published the first results of a longitudinal study on five online marketplaces, to analyze the evolution of the individual ecosystems.

The scientific contribution of the research project described in this document is manifold. First of all, the investigation, visualization and exploration of new ecosystems further fosters the body of knowledge of the research domain of software ecosystems. Second, so far limited attempts have been made to visualize and compare proprietary platform ecosystems. This causes a discrepancy, since forces and mechanisms at play in closed software ecosystems differ from those that affect open ecosystems (Economides & Katsamakas, 2006). Third, quantifying ecosystem characteristics by means of network metrics aid in the characterization of ecosystems and the comparison thereof by providing means to formalize the comparison process. The comparison of platform ecosystems helps in uncovering the factors that make ecosystems differ from each other, which in turn benefits the characterization of software ecosystems, a main research challenge described by Jansen, Finkelstein, and Brinkkemper (2009), Barbosa and Alves (2011) and Hanssen and Dyba (2012). Last, gaining insight into mechanisms at play in an ecosystem can provide directions and indicators that can be used to evaluate the effectiveness of ecosystem orchestration mechanisms through longitudinal studies.

1.3.2 Practical Relevance

As elaborated in previously conducted network visualization studies, in-depth ecosystem analysis and assessment can wield an array of benefits for practitioners. Iyer et al. (2006), for example, suggest that managers of software companies should implement a continuous ecosystem monitoring process. Iyer et al. (2006) argue that managers should create a "network scorecard' for their company, to evaluate their position in the ecosystem and the influence that actions of other actors have on the ecosystem as a whole. To establish such a scorecard, however, the characteristics of an ecosystem should be quantifiable and measureable, in order to provide adequate analysis. Basole (2009) and Den Hartigh et al. (2006), also demonstrate how network visualization can lead to advanced insights and evaluation of network performance of individual partners in the ecosystem. In the context of open ecosystems, Kabbedijk and Jansen (2011) could provide the Ruby on Rails community management with strategic advise after network analysis revealed that 90% of the activity in the community was generated by only 10% of the developers. To aid managers in performing ecosystem analysis for their own ecosystem, this research project provides deeper insight into factors that shape or influence platform ecosystems.

Furthermore, this research project provides insight into software ecosystem orchestration, and relates software platform management to ecosystem management. As a consequence, increased insight in the factors that shape platform ecosystems can assist software companies in making better informed decisions with regard to their own platform and consecutive ecosystem. In turn, this could benefit the performance of a software organization, by better aligning platform strategies with software ecosystem goals.

1.4 Document Structure

This thesis continues with a description of the research approach in Chapter 2. This chapter provides details on the research methods used to collect the findings that are presented further on this thesis. In addition, this chapter addresses the threats to validity as anticipated in before the execution of the research project.

The theoretical part of this thesis begins with a summary of findings taken from existing scientific literature in Chapter 3. This chapter introduces principal concepts that are central to platforms and ecosystems. The chapter also provides an overview of methods that can be used to visualize platform ecosystems and the interactions that take place within such ecosystems, alongside their advantages and drawbacks. Hereafter, Chapter 4 crafts and formalizes network metrics that can be used to analyze platform ecosystems.

The empirical results obtained from the data collection process are described in Chapter 5. This chapter describes the network topology of four platform ecosystems, being the cloud based office suites Google Apps and Microsoft Office365, and the web browsers Google Chrome and Internet Explorer. Each ecosystem is visualized by means of network graphs, and calibrated through the network metrics that have been formalized in the preceding chapter. The description of the platform ecosystems, is then input for the analysis and ecosystem comparison presented in Chapter 6.

The analysis is followed by a discussion in Chapter 7. The discussion contains a description of the data collection method validation and an elaboration upon reservations that need to be made when interpreting the results presented in this thesis. The discussion is accompanied by a conclusion, a summary of the findings presented throughout this thesis and suggestions for future research.

Chapter 2

Research Approach

In performing this research a combination of a literature review and case studies was carried out. A literature review was performed to provide theoretical background on the topics of software ecosystems, software platform management and the application of network analysis in the context of interconnected businesses or interfirm networks. Furthermore, empirical findings at a later stage of this research project were compared with findings from previous literature to enhance internal and external validity of the presented findings. Next, four platform ecosystems were selected for a theory building multiple case study (Eisenhardt, 1989; Eisenhardt & Graebner, 2007; Yin, 2009), from which propositions were inducted and developed. This approach was chosen because of the relative novelty of the research domain. To provide for improved triangulation of evidence, two sets of comparable ecosystems of two platform owners were selected as subject of study.

All consecutive research steps and their corresponding (intermediate) deliverables are shown in Figure 2.1, that was created employing modeling techniques adopted from the research domain of method engineering (Brinkkemper, 1996). Figure 2.1 is composed of a simplified UML activity diagram and a UML concept diagram (i.e. a simplified UML class diagram) in accordance with the meta-modeling standards defined by Van de Weerd and Brinkkemper (2008). The activity diagram describes and interrelates the main steps in the research process while the UML concept diagram interrelates the deliverables that are produced as a result of a research step. Combining the two diagrams – by connecting the activities and resulting deliverables with dashed arrows – leads to the creation of a process-deliverable diagram (PDD), a common aspect of method description in the field of method engineering (Brinkkemper, 1996). The activities and concepts shown in Figure 2.1 are described and defined in respective tables. The activities are described in

Table 2.1 and Table 2.2, in which the concepts associated with an activity are included in capitals. The concepts then, are defined in Table 2.3.

TABLE 2.1: Activity table for the preparation activities in the research approach process-deliverable diagram

Identify research gap		Based on an exploration of existing literature, a research gap is identified. This research gap is translated into a research problem and formulated in the PROBLEM STATEMENT.
Formulate research questions		To address, operationalize and scope the identified research problem, one or more RESEARCH QUESTIONs are formulated.
Conduct focused literature review		To provide the THEORETICAL BACKGROUND of a research, findings from existing literature are elaborated and summarized as a fundament for the empirical part of a research project.
Formulate hypotheses		To operationalize construct measurement, a HYPOTHESIS has to be formulated that is testable by means of empirical evidence.
Prepare case studies	Formulate case study selection criteria	To increase validity, CASE STUDY SELECTION CRITERIA are formulated that a potential case study subject has to adhere to.
	Select case studies	Based on the CASE STUDY SELECTION CRITERIA, select four case studies that will be subject of study in this research project.
	Create case study protocol	To ensure that the same approach is taken in conducting each individual case study, all empirical research steps and measurements are predefined in a CASE STUDY PROTOCOL.

Table 2.2: Activity table for the execution and analysis activities in the research approach process-deliverable diagram

Collect case study data	Perform external document study	External documents available for each individual case study subjects such as portals, websites and technical documentation are analyzed, the findings of which are described as a DOCUMENT STUDY.
	Identify actors that are part of the ecosystem	By means of a web crawler, all ACTORs are identified and their metadata is stored in a database. ACTORs are assembled from extension markets and included in a LIST OF ACTORS.
	Identify relationships among actors	Interfirm relationships between ACTORs that are part of the LIST OF ACTORS are identified by querying CrunchBase and assessing the company website and newsfeeds of every single ACTOR. Accordingly, all ECOSYSTEM RELATIONSHIPs are documented.
	Validate completeness and accuracy of data	To verify the accuracy and completeness of the LIST OF ACTORS and ECOSYSTEM RELATIONSHIPs, Google Apps vendors are contacted. The vendors are asked whether the partner lists compiled for their company were complete.
Analyze case study data	Perform single case analysis	Empirical data per case study are analyzed by assembling the DOCUMENT STUDY, LIST OF ACTORS and ECOSYSTEM RELATIONSHIPs into a CASE STUDY REPORT, that contains all relevant data on one ecosystem.
	Search for cross-case patterns	Groups or pairs of case studies are compared to identify similarities and differences among them, that are subsequently formulated as a CROSS-CASE PATTERN.
	Formulate initial theory	One or more CROSS-CASE PATTERNS are translated into an INITIAL THEORY, that explains an observed phenomenon.
	Test initial theory	An in-depth analysis of theory and empirical data provides for confirming or disproving the INITIAL THEORY.
	Enfold literature around theory	To raise the theoretical level and increase the external validity of a THEORY, existing literature is examined to provide for confirming and contradicting statements.
	Reach closure of theory	At the point of theoretical saturation (e.g. when new iterations do not provide significant improvements), the INITIAL THEORY is formulated into a THEORY.

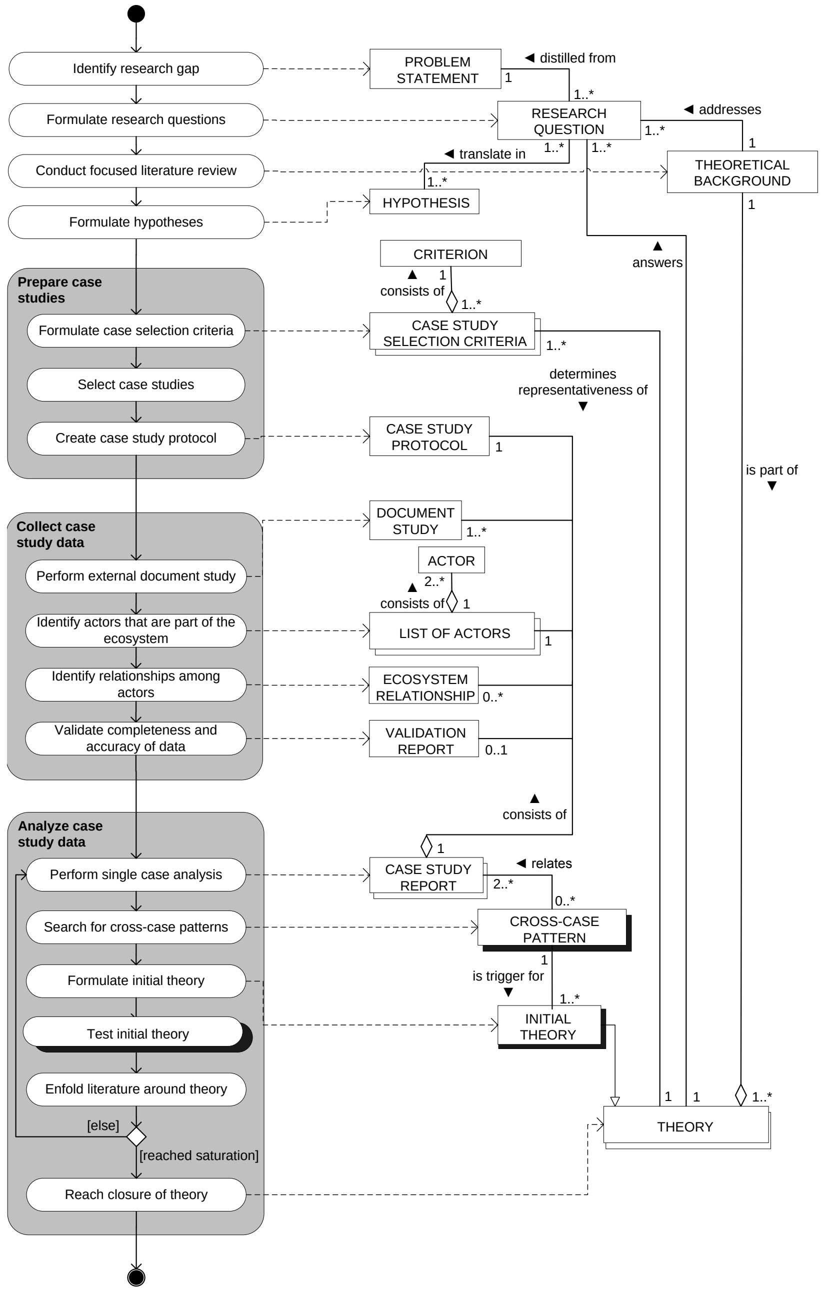

Figure 2.1: The research approach described by means of a process-deliverable diagram

Table 2.3: Concept table for the research approach process-deliverable diagram

Concept	Description
PROBLEM STATEMENT	A definition of the research problem that is addressed in a research project (Eisenhardt, 1989).
RESEARCH QUESTION	A question that scopes and defined the research objectives of a research project at hand (Eisenhardt, 1989).
HYPOTHESIS	A quantifiable and testable specification of a requirement for a case study (Kitchenham & Pickard, 1998).
THEORETICAL BACKGROUND	An elaboration of the findings from previous research on a certain topic.
CRITERION	A requirement a potential case study subject has to adhere to (Yin, 2009).
CASE STUDY SELECTION CRITERIA	A list with multiple CRETIONs a potential case study subject has to adhere to in order to be selected as a final case study subject (Yin, 2009).
CASE STUDY PROTOCOL	An outline of the case study process, predefining all the necessary research steps and measurements together with a documentation of their relation to the research goal or (sub) research question (Yin, 2009).
DOCUMENT STUDY	The results of an analysis of internal or external documentation available on a case study subject (Bowen, 2009).
ACTOR	An entity that shapes, orchestrates, participates in, or contributes to a software ecosystem depending on its role, type and purpose (Van Angeren, Blijleven, et al., 2013).
LIST OF ACTORS	An overview of all ACTORs that are part of a software ecosystem.
ECOSYSTEM RELATIONSHIP	A connection between two or more actors, characterized by the flow of products, services, money and intellectual property (Van Angeren, Blijleven, et al., 2013).
VALIDATION REPORT	A document with findings obtained from the data validation process (Yin, 2009).
CASE STUDY REPORT	A document that elaborates on the main findings from a case study alongside references to the empirical material used in the process (Yin, 2009).
CROSS-CASE PATTERN	Similarities or differences between pairs or groups of case studies as observed or uncovered by a researcher (Eisenhardt, 1989).
INITIAL THEORY	A description of a one or more CROSS-CASE PATTERNs (Eisenhardt, 1989).
THEORY	A set of interrelated constructs, definitions and propositions that present a systematic view of a phenomenon by specifying relations among variables, with the purpose of explaining and predicting the phenomenon (Kerlinger, 1986).

2.1 Literature Review

A scoped literature review (Arksey & O'Malley, 2005) provided the theoretical background of this research, and means to formalize network metrics for ecosystem analysis (first sub research question). Contrary to a systematic literature review – that is employed to provide an encompassing overview of research carried out in a specific domain – a scoped literature review enables researchers in providing a fundament for consecutive steps in the remainder of a research project. According to Arksey and O'Malley (2005), scoped literature reviews can be used to build a theoretical background for a research project, to identify research and knowledge gaps, or can be part of a mixed-method research (Dellinger & Leech, 2007). To retrieve relevant sources, the collection process consisted of a combination of automated and manual search queries for English peer-refereed scientific contributions. Automated searches were performed on the scientific databases of ACM, Emerald, IEEE, ScienceDirect, Springer and Wiley. The following keywords were combined into search strings:

Keywords: *"business ecosystem" AND/OR "software ecosystem" AND/OR "digital business ecosystem" AND/OR "digital ecosystem" AND/OR "platform ecosystem" AND/OR "business relationship" AND/OR "interfirm relationship" AND/OR "platform leadership" AND/OR "software platform" AND/OR "industry platform", "proprietary platform" AND/OR "platform openness" AND/OR "platform architecture" AND/OR "platform strategy" AND/OR "ecosystem strategy" AND/OR "ecosystem analysis" AND/OR "network analysis" AND/OR "graph theory"*

Important in conducting literature reviews, is the predefinition of inclusion and exclusion criteria for the studies found through both automated and manual searches. Accordingly, the following inclusion and resulting exclusion criteria were defined:

1. Studies are written in English.
2. Studies are peer-reviewed.
3. Studies are included disregarding their publication date.
4. Studies that are inaccessible are not included.

To retrieve additional scientific literature, bibliographies of selected articles were reviewed. Relevant sources were considered based on their title, keywords and abstract.

2.2 Case Studies

The empirical part of this research addressed the remaining three sub research questions and consisted of four theory building case studies. Two platforms per case company were selected as subject of study. To ensure research rigor, the case study process has been based on the theory building case study guidelines as defined by Eisenhardt (1989) and Yin (2009). Eisenhardt (1989), defined a set of iterative research steps when building theory from empirical research, covering the process from the induction of a theory to reaching scientific closure: the stage at which theoretical closure is reached and where a new theory is established. Apart from defining the research questions the process entails the following steps:

1. Selecting case studies
2. Crafting research instruments and protocols
3. Data collection
4. Data analysis
5. Shaping initial theory
6. Enfolding literature
7. Reaching closure

Each of the research steps defined by Eisenhardt (1989) will be elaborated in the remaining subsections of this section.

2.2.1 Case Study Selection

Contrary to survey research in which large and random populations are subject of study, case study selection is an important step in theory building case study research. Because of the smaller number of case studies being carried out, randomly selecting case studies is undesired (Eisenhardt, 1989). Rather, cases are selected to suit the research at hand, for example, to replicate existing case studies or to apply a theory within a new domain. Furthermore, case study selection benefits external validity, and thus the degree of generalizability of the findings from empirical research by ensuring that the case study sample is both enough representative though diverse at the same time (Yin, 2009).

To provide for triangulation of evidence, this research project addressed two platforms per selected case company. Google and Microsoft were selected as case companies due to the overlap in their platform portfolio. When selecting platforms, a long list with platforms of Microsoft and Google was created. The platform portfolios of Google and Microsoft were compared to identify common elements for both companies (e.g. Microsoft and Google among others both have a mobile operating system, a web browser, an enterprise application suite and business application stores). The long list with common platforms has been included in Table 2.4.

Table 2.4: Long list of overlapping platforms in the platform portfolios of Google and Microsoft

Platform type	**Google**	**Microsoft**
Academic search engine	Google Scholar	Microsoft Academic Search
Advertisement	Google AdWords	Bing Ads
Enterprise application marketplace	Google Apps Marketplace	Microsoft Pinpoint
Cloud (business) office application suite	Google Apps	Microsoft Office365
Cloud file hosting	Google Drive	SkyDrive
Collaborative office suite	Google Docs	Microsoft Live Office
Mail	Gmail	Outlook.com
Mobile operating system	Android	Windows Phone
Operating system	Chrome OS	Windows
Photo organization and editing	Picasa	Windows Photo Gallery
Search engine	Google Search	Bing
Social network	Google+	So.cl
Web browser	Chrome	Internet Explorer
Web mapping	Google Maps	Bing Maps

Out of the overlapping elements in the application portfolios, two platforms per company were selected based on the following selection criteria:

1. **The platform of interest is available in multiple regions:** The application portfolio of the case companies consist of a large number of elements, part of which are only available in certain parts of the world. As an example, Google Books – a mobile e-reading application and content store – is only available to users in the United Kingdom, Northern America and Australia. To enhance the representativeness of the performed ecosystem study, platforms that are not available globally were not considered as potential case study subject.

2. **The platform and surrounding ecosystem of interest are documented on a dedicated portal or website:** Transparency is a pivotal criterion in performing

case study research (Yin, 2009). Subsequently, only platforms on which extensive publicly accessible documentation is available were considered for inclusion.

3. **Data and metadata about actors that are part of the platform ecosystem must be retrievable by means of a web crawler:** Since part of the data collection process was automated by means of a crawler, data and metadata about complementors had to be retrievable with a web crawler from an app store or extension market. Suitable for scraping are lists of applications, plug-ins and other extensions. The minimal available data and metadata encompasses application names, company names of complementors and websites from complementors.

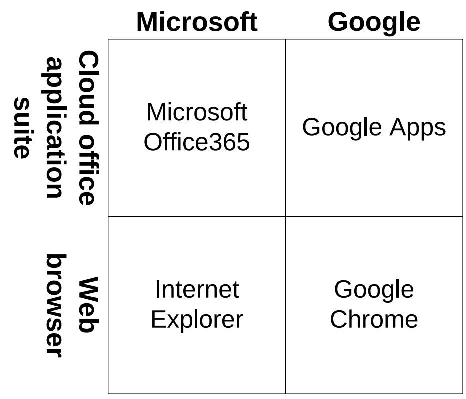

Figure 2.2: Matrix categorization of case study subjects

Four case study subjects were selected from the product portfolio of Microsoft and Google. In accordance with the formulated inclusion criteria, the web browsers Internet Explorer and Google Chrome, and the cloud based business office suites Microsoft Office365 and Google Apps have been selected as subject of study. The matrix classification in Figure 2.2 provides an overview of the selected platform ecosystems.

2.2.2 Case Study Protocol and Data Collection

According to Eisenhardt (1989), theory building researches are typically conducted by combining multiple data collection methods, as it provides for triangulation of evidence, a variety of insights and perspectives. Consequently, in collecting data for each individual platform, multiple data gathering techniques were used. These techniques included an external document study, app store data extraction by means of a web crawler and manual identification of business relationships. When utilizing multiple data collection methods, the definition of a case study protocol becomes a necessity (Yin, 2009). A case study protocol predefines all necessary research steps and measurements. The definition of such a protocol ensures that a similar approach is taken in conducting each individual case study, which in turn increases validity and reliability of the findings deducted from this research. The case study protocol employed in this research is based on the case study protocol template described by Brereton, Kitchenham, Budgen, and Li (2008), and is included in Appendix A.

The data collection process for each individual case study started out with a study of the documentation provided by the case company on the platforms of interest. This documentation consisted of information found on dedicated portals, websites, or in technical documentation. It provides information on existing extension mechanisms for a platform, such as APIs, SDKs and plug-in architectures. Furthermore, the document study aided in the identification of the current initiatives employed by a platform owner to orchestrate the ecosystem, such as partnership models or certification programs.

Automated data extraction provided for an overview of the complementors that currently develop one or more applications for one of the platforms. As the focus of this research concerns software complements, specialized services were ignored, meaning that for Office365 and Google Apps only products listed under the category *'applications'* or *'products'* were considered. A crawler was used to retrieve application specific metadata from the online marketplace that showcase complements for the platform. This application specific metadata includes information about the vendor, pricing model, certification status, partner information, and reviews. The crawler used in this research leans on the architecture of a previous initiative by Burkard et al. (2011), who in 2010 and 2011 retrieved application specific metadata from five online marketplaces on a weekly basis. The crawler is Java-based and has a two layer architecture. The universal layer is responsible for all data handling. The second layer provides data retrieval from the marketplace.

All application specific metadata for each platform was retrieved in two stages. First, the crawler traversed the entire list of complements in the online marketplace over multiple iterations to the point at which no new applications were found. For some marketplaces this is one listing, while for other marketplaces all available categories had to be traversed. Second, all application specific metadata could be read-in by Java through their respective URLs, in the case of non-AJAX based marketplaces. The webpages retrieved through the URLs were converted into Document-Object-Models with an open-source tool called NekoHTML[1], as this simplified the parsing of data. Afterwards, all data was stored in a central case study MySQL database (Yin, 2009). In case of an AJAX marketplace, Watij[2] had be used to be able to retrieve data, in before conversion into

[1]http://www.nekohtml.sourceforge.net
[2]http://www.watij.com

Document-Object-Models took place. For Office365, this process was repeated over multiple iterations, as Microsoft uses separate marketplaces for different countries.

Contrary to the application specific metadata, the formal relationships between complementors could not be retrieved automatically. However, formal relationships such as alliances, joint ventures, technological partnerships, distribution partnerships and participation in partnership models could be identified from multiple sources. Among these sources are company websites, partner portals, company news feeds, and proprietary or openly accessible databases (Basole, 2009; Iyer et al., 2006). Accordingly, all complementors were identified by means of an SQL query on the database with vendor data followed by manual verification, as some companies use multiple aliases in one marketplace. The company website of each vendor was then inspected to identify the business relationships they maintain with other actors in the network. In addition, CrunchBase[3] was queried to identify additional interfirm relationships that exist in the ecosystem. Herein, relationships are regarded as undirected bonded-ties (Hanneman & Riddle, 2005), meaning that if NetSuite lists a relationship with Box, Box is also considered to be in a business relationships with NetSuite, even if Box omitted to list NetSuite as a partner.

The identified relationships were maintained in an adjacency matrix, created with a tool called UCINET[4]. An adjacency matrix is a matrix in which for every pair of actors (e.g. an adjacency pair), the presence or absence of a relationship can be indicated (Hanneman & Riddle, 2005). In the adjacency matrix, **'1'** was used to indicate the existence of a partnership, and **'0'** indicated that two actors do not have a relationship. Table 2.5 shows part of the adjacency matrix for the Google Apps ecosystem to illustrates how relationships were archived.

Table 2.5: Adjacency matrix for a random sample of the Google Apps ecosystem

	Google	**Zoho Corp.**	**SaaSt**	**Top-Solutions**	**Cloud-Work**	**ekto-sym**	**florey-soft**
Google	-	1	1	1	1	1	1
Zoho Corp.	1	-	0	1	0	0	0
SaaSt	1	0	-	0	0	0	0
TopSolutions	1	1	0	-	0	0	0
CloudWork	1	0	0	0	-	0	0
ektosym	1	0	0	0	0	-	1
floreysoft	1	0	0	0	0	1	-

Because of its compatibility with UCINET, Gephi[5] (Bastian, Heymann, & Jacomy,

[3]http://www.crunchbase.com
[4]http://www.analytictech.com/ucinet
[5]http://www.gephi.org

2009) was used to visualize the ecosystem by means of network graphs. The structural properties of the network were computed with either UCINET or Gephi.

2.2.3 Data Analysis

Typically, in case study research there is an overlap between data gathering and data analysis. In this research, for example, the results of the document study did influence the subjects of data extraction. In theory building case study research, a distinction is made between five elementary steps of data analysis: (1) single case data analysis, (2) cross-case data analysis, (3) shaping initial theory, (4) enfolding literature and (5) reaching closure (Eisenhardt, 1989). As shown in Figure 2.3, this process is highly iterative, meaning the first four steps in the process are performed over multiple iterations. The iteration over the four process steps in data analysis at the end leads to closure: the establishment and description of a new theory.

Within-case data analysis involves processing and interpreting data for an individual case study (Eisenhardt, 1989). Each of the platform ecosystems was visualized by means of a network graph and principal actors were identified (second sub research question). Furthermore, important findings from the document study have been presented in a case study report. Ensuring uniformity, each case study report consisted of the following elements:

- **Platform description:** A description and high level overview of the platform.
- **Adjacency matrix:** A table that contains an overview of all existing relationships between pairs of actors in the ecosystem.
- **Network graph:** A graphical representation of the complementor ecosystem that is constituted of all complementors and the relationships among them.
- **Actor classification:** A classification of important actors in the ecosystem based on their role in the network. The employed actor categorization is based on a priori defined network metrics that have been taken from existing scientific literature.
- **Network structure:** A characterization of the complementor ecosystem based on network metrics, which were defined a priori based on findings from literature.

Consecutively, the second step in the analysis process was the search for cross-case patterns, where observations are made about commonalities and differences among the four different case studies. Within this research project, cross-case analysis has been performed in two successive phases – ensuring that only one variable differs while the

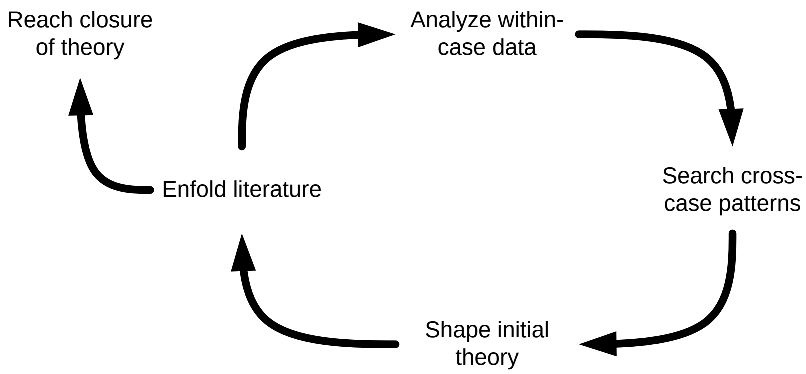

FIGURE 2.3: The process of data analysis in theory building case study research

other potential variable remains controlled. First, a comparison was made between the two platform ecosystems of each case company (third sub research question). Afterwards, the similar platforms and their respective ecosystems were compared to uncover the main similarities and differences (fourth sub research question). This comparison was performed by means of observation, visual examination and statistical comparison.

Related to cross-case analysis is the process of shaping one or more initial theories based on the cross-case observations. The observation of certain events, similarities or differences function as a trigger in the process of formulating an initial theory, which is comparable to formulating hypotheses in quantitative research (Eisenhardt, 1989). The initial theories then, were tested based on an in-depth study of the case data or existing scientific literature. When enfolding literature, the goal was to identify corresponding or contradicting statements described in previous literature compared to the content of the initial theory. Enfolding literature raises the theoretical level of the empirical findings – a common evaluation mechanism in case study research (Yin, 2009) – and aids in increasing external validity. When saturation (e.g. the point at which it is highly unlikely that advances over new iterations will result in new insights) has been reached, no new iterations were performed. Instead, closure of theory has been reached.

2.2.4 Validation

As the data collection approach described in the preceding subsections is novel and has not been applied to visualize proprietary platform ecosystems before, a validation step was implemented. The purpose of the validation step was to verify if the method to identify business relationships among complementors is complete and accurate. This is a necessary step as this research depends on manual data retrieval from proprietary sources.

Thirty-five of the largest vendors (i.e. that develop the most applications) in the Google Apps ecosystem were contacted through email. The complementors were presented with preliminary findings about the Google Apps ecosystem and a list with their partners as identified during the data collection process. Complementors were asked to confirm whether the partner list compiled for their company was complete and accurate, and whether they feel that the network graph of the Google Apps ecosystem provides an exhaustive overview of the ecosystem and its interactions.

2.2.5 Plan Validity

Case study research is an example of applied research, that is recurrently applied in the software industry (Host & Runeson, 2007). Contrary to laboratory settings – in which the environment can be manipulated in such a way that the influence of mediating variables can be omitted – case study research takes place in real-life settings where the environment is uncontrolled. Subsequently, case study designs must limit the mitigating influence of the environment or other contextual factors as much as possible (Yin, 2009). To address the necessity of evaluating case study designs, the case study protocol included in Appendix A has been created in accordance with the case study research checklist by Host and Runeson (2007) and Runeson and Host (2009), which addresses the accuracy, completeness and rigor of the case study research project to be executed. Yin (2009) translates the quality requirement for case study designs into four conditions that have to be addressed to maximize the quality and value of case study research, which are elaborated upon in detail in the remainder of this subsection:

1. Construct validity
2. Internal validity
3. External validity
4. Reliability

Construct validity is involved with operational measures, and whether they are suitable to measure theoretical constructs (Yin, 2009). For example, the theoretical construct 'time-lapse' can be measured by counting seconds.

As this research addresses the factors that shape the network structure of a complementor ecosystem, network metrics and scores deducted from matrices were chosen as subject of analysis. To increase construct validity, a list of network metrics and characteristics was distilled from previous research and translated into the context of the interconnected software business. Construct validity is also pivotal in reconstructing the relationships that exist within the ecosystem. To increase its accuracy and completeness, triangulation of evidence is strived for by combining sources such as company websites and business databases in the data collection process. Another way to increase construct validity is to have case study participants evaluate draft case study reports. As this research depends solely on externally available resources, the draft case reports were compared with similar previously conducted studies (Basole, 2009; Iyer, 2012; Iyer et al., 2006; Kabbedijk & Jansen, 2011; Madey, Freeh, & Tynan, 2004) to verify its completeness. Additionally, Google Apps vendors were asked to validate preliminary findings.

Internal validity regards casual relationships, and is involved with ensuring that the proposition studied in a case study is indeed a correct proposition that omits the influence of other variables (Yin, 2009).

According to Yin (2009), the preferred strategy for ensuring internal validity is reusing propositions from existing scientific literature, a strategy that has been adopted for this research project. As this thesis evaluates the influence of both the type of platform and the firm ecosystem on the network structure of platform ecosystems, each variable has been addressed in a separate sub research question. Doing so, the other variable can be controlled, preventing it from having mediating effects on the case study results and analysis.

External validity addresses the generalizability of case study findings outside the context of a single case study (Yin, 2009).

To benefit generalizability, a multiple case study design was chosen, from which unique case studies were deliberately excluded. Accordingly, two platform owners were selected, and out of the total dataset two sets of comparable platforms were chosen as subject of study, which according to Hanneman and Riddle (2005) can be considered as sampling. After the case study analysis, contradicting and confirming findings from previous literature were used to embed the empirical findings within the current scientific landscape – a common evaluation method in case study research (Eisenhardt, 1989).

Reliability of case study research is involved with replicability of a case study research, to ensure that if a later researcher follows the procedures as described in the original research the researcher should be capable of conducting the case study research in a similar fashion (Yin, 2009).

To increase the reliability of the research project described in this thesis, Appendix A contains the case study protocol that prescribes all research steps taken throughout this research. A central case study database has been employed to store all relevant case study data. Accordingly, all findings have been cross-referenced to increase transparency and traceability. Furthermore, pivotal intermediate deliverables of this research project can be found in the Appendices of this thesis to benefit transparency.

Chapter 3

Theoretical Background

In a retrospective review of business and ecosystem literature, Hearn and Pace (2005) identify five shifts that characterize the ecological metaphor of interconnected businesses: (1) the shift from thinking about consumers to co-creators of value, (2) the shift from thinking about value chains to value networks, (3) the shift from thinking about product value to network value, (4) the shift from thinking about simple co-operation and competition to complex co-opetition, and (5) the shift from thinking about individual firm strategies to strategy in relation to value ecologies. As this section provides an overview of the elementary theoretical background for this research, these five shifts are illustrated both from the perspective of platform-based thinking and the ecosystem metaphor. This section regards these topics as separate entities, sheds light on the analogies that exist between them and elaborates on existing means that have been used to analyze platform ecosystems.

3.1 Platforms

From the 1990s onward, scholars have been directing attention towards platforms. Nowadays, platforms are subject of study for academia from domains such as economics, software engineering and business strategy. While research attention for platforms has emerged over recent years, Baldwin and Woodard (2009) trace the societal use of the word 'platform' back to the sixteenth century, where a platform was "a raised level surface on which people and things can stand, usually a discrete structure intended for a particular activity and operation" (p. 2). During recent years, scholars have been concerned with the provision of formal and applicable definitions for platforms, both inside and outside of the software industry. Examples of definitions introduced in scientific literature include the definition by Robertson and Ulrich (1998) who define a platform as "a collection of assets (e.g. components, processes, knowledge, people and relationships) that are shared by a set of products" (p. 2), the definition by Bresnahan and Greenstein (1999) who regard a platform as "a bundle of standard components around which buyers and sellers coordinate" (p. 4) and Baldwin and Woodard (2009) who define a platform as "a set of stable components that supports variety and evolvability in a system by constraining the linkages among the other components" (p. 1). While the definitions reside from different domains and thus take different perspectives on platforms, common elements can be distilled. All three definitions touch upon (1) a certain core of stable components and (2) reuse or extension by other components or parties. As this thesis puts emphasis on the ecosystem of complementors around the actual platform, the described definitions were found obsolete. Instead, in this thesis a platform is defined as follows (Gawer & Cusumano, 2002):

Platform: A foundation technology or set of components used beyond a single firm that brings together multiple parties for a common purpose or recurring problem.

While platforms have become an increasingly popular research subject, various scholars take differing perspectives on platforms. Attention for platforms does not solely stem from the software industry, rather platforms are found all throughout modern-day manufacturing and production. Therefore, the remainder of this section elaborates on different types of platforms, their common characteristics and architecture, and the importance of platform strategy to foster the position of the platform owner.

3.1.1 A Typology of Platforms

In modern-day literature, academia refer to all kinds of physical or intangible products (e.g. software products) as platforms, that among others include credit cards, ATMs, shopping malls, personal computers, mobile phones, enterprise resource planning software and (mobile) operating systems (Baldwin & Woodard, 2009; Bosch, 2009; Gawer & Cusumano, 2002; Gawer, 2009a; Jansen & Cusumano, 2012). Platforms can be used for production within one firm such as the Sony Walkman (Sanderson & Uzumeri, 1995), but can also be considered an industry-wide standard such as Microsoft Windows (Gawer & Cusumano, 2002; Gawer, 2009a). Since attention for platforms stems from different academic fields including new product development, technology strategy and economics, scholars have been involved with developing typologies of platforms.

In a two-dimensional typology, Bosch (2009) makes a distinction between desktop platforms, web platforms and mobile platforms. Each platform category then, is decomposed in operating systems, applications, and end-user programming. End-user programming refers to domain-specific programming languages or tools with which an end-user can generate applications themselves. Operating systems are regarded as domain independent platforms, whereas applications are software products such as Facebook, that have been opened up at a later stage by means of an API to facilitate third-party extension (Bosch, 2009). Perhaps even more functional in their nature are (application) development platforms such as Eclipse or Mendix, which serve as an environment to develop or derive products.

A less functionality-oriented typology of platforms comes from Gawer (2009a), who distinguishes four types of platforms:

1. Internal platforms
2. Supply chain platforms
3. Industry platforms
4. Two-sided markets

TABLE 3.1: A typology of platforms

Factor	**Internal platform**	**Supply chain platform**	**Industry platform**	**two-sided market**
Context	Within an organization	Within a supply chain	Industry or ecosystem	Industry or ecosystem
# of participants	One firm	Several firms within a supply chain	Several firms that do not necessarily buy or sell from each other, whose products or services function together as a technological system	Several firms who transact with each other, facilitated by an intermediary platform
Example	Walkman, Microsoft Office	Acer personal computers	Microsoft Windows, Android	Apple AppStore, Google Apps Marketplace

Among others, the platforms differ in relation to the context in which they are used and the number of participants that contribute to the platform or its diversification. Table 3.1 summarizes part of the typology of Gawer (2009a) (e.g. only the characteristics that are relevant for this thesis), and furthermore includes illustrative examples of each platform.

According to Gawer (2009a), attention for platforms in an early stage predominantly originated from the context of new product development. From the perspective of new product development, the use of platforms has been observed and elaborated within a single firm. Gawer (2009a), refers to these platforms as internal platforms, whereas others such as Meyer and Lehnerd (1997), Robertson and Ulrich (1998), McGrath (2000) and M. Martin and Ishii (2002) refer to internal platforms as product platforms. In scientific literature, various definitions for product platforms can be found, including the early definition of Wheelwright and Clark (1992) who regard a product platform as "products that meet the needs of a core group of customers that are designed for easy modification into derivatives through the addition, substitution or removal of features" (p. 5) and the abstract definition by Krishan and Gupta (2001) that define a product platform as "component or subsystem assets shared across a family of products" (p.1). As the former definitions are either abstract or originate from the early days of platform research, this thesis adopts the definition by Muffato and Roveda (2002, p. 3) which is an enhancement of the original definition by Meyer and Lehnerd (1997):

Internal platform: A set of subsystems and interfaces to form a common structure from which a stream derivative products can be efficiently developed and produced.

As can be noted from the definition, internal platforms encompass a set of components or functionality that are used across multiple products. According to Pine (1993), internal platforms are the solution to the facilitation of mass customization, which ensures maintaining benefits from economies of scale. In addition, researchers mention reduced time-to-market (Wheelwright & Clark, 1992), profit maximization for product families (Suh, De Weck, & Chang, 2007) and achieving familiarity in user experience between different products (Bosch, 2002) as key triggers for the embracement of internal platforms. Internal platforms are used in various manufacturing industries, such as the automotive industry where platforms are used to build multiple types of cars on the same platform (e.g. basis) (Suh et al., 2007). Platforms are also used in manufacturing airplanes (Gawer, 2009a) or electronic devices (McGrath, 2000). In the software industry, the use of internal platforms has been adopted in the creation of software product lines. Bosch (2002) describes a software product line as an intra-organizational software reuse approach. A software product line consists of an internal platform, and separate products that reuse part of the functionality of the platform and extend it with additional functionality.

As the popularity and adoption of internal platforms in industries increased, new initiatives were introduced to enable the use of platforms on an inter-organizational level. Organizations faced high variety in parts or intermediate products that were produced in their supply chain, because the degree of standardization was found to be low (Gawer, 2009a). Accordingly, supply chain platforms were introduced and adopted in industries such as the automotive industry (Zirpoli & Caputo, 2002) and high-tech industry (Hammel, Phelps, & Kuettner, 2002). In accordance with their application in practice, supply chain platforms are defined as follows (Gawer, 2009a, p. 8):

Supply chain platform: A set of subsystems and interfaces that forms a common structure from which a stream of derivative products can be efficiently developed and produced by partners along a supply chain.

Supply chain platforms are similar to internal platforms in their architecture (Gawer, 2009a). Equal to internal platforms, the architecture of supply chain platforms is based on reuse of components that are part of the modular system of the platform. Furthermore, only one organization governs and thrives innovation for or around the platform. This, either is an assembly firm that outsources part of the development to its suppliers in accordance with predefined standards or a firm that shares ownership with another company as part of an alliance (Gawer, 2009a).

Contrary to internal and supply chain platforms – that thrive on the contribution of only one or a small group of stakeholders – industry platforms and two-sided markets influence groups of actors. Common examples of industry platforms are Microsoft Windows, the Apple iPhone and its operating system iOS, and social media such as Facebook and Twitter (Baldwin & Woodard, 2009; Bosch, 2009; Gawer & Cusumano, 2002; Gawer, 2009a), reflecting that industry platforms are predominantly found in the software industry. Gawer (2009a, p. 10) defines an industry platform as follows:

Industry platform: 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.

As noted from the definition, the platform itself only provides a foundation upon which complements are build by third parties. These parties together are referred to as the industry (Gawer & Cusumano, 2002; Gawer, 2009a) or ecosystem (Gawer, 2009b; Iansiti & Levien, 2004b; Jansen, Brinkkemper, & Finkelstein, 2009). The strong tie with the ecosystem is especially emphasized in the definition of a platform by Iansiti and Levien (2004a), who regard a platform as "a set of solutions to a problem that is made available to the members of the ecosystem through a set of access points or interfaces" (p. 148). Typical for industry platforms is that the success and value of the platform to a large extent is dependent on the variety of complements that surround the platform, and the innovation speed of the respective complementors (Gawer, 2009a; Popp & Meyer, 2010). For example, the Microsoft Windows operating system itself exhibits limited functionality, solely aiding users in the performance of elementary tasks with desktops, laptops or tablet computers. Third parties then, develop applications such as games and administrative software that address the wishes of a new or existing customer base. Similarly, SAP or Salesforce depend on partners to build domain-specific extensions to make their enterprise applications functionality usable in specific market segments. Accordingly, (application) development platforms are not considered industry platforms, because their derivatives do not represent any complementary value to the platform as such.

Throughout the years, different names have been introduced for platform owners in the context of industry platforms, including platform leaders as introduced by Gawer and Cusumano (2002), platform sponsors (Eisenmann, Parker, & Van Alstyne, 2009) or keystones, as platform owners are termed by Iansiti and Levien (2004a). While terminology differs, there is consensus among researchers (Baldwin & Woodard, 2009; Gawer & Cusumano, 2002) that benefits from network effects are the main driver for establishing industry platforms. Additionally, the position of the platform owner has changed compared to internal or supply chain platforms. In managing industry platforms, the platform owner no longer is the sole actor responsible for innovation, standard definitions and governance, rather it heavily depends on the initiatives developed by the complementors that surround the platform (Gawer, 2009a; Jansen & Cusumano, 2012).

Two-sided markets are identified as the last pillar of the platform typology by Gawer (2009a). A recurrently named example of a two-sided market is a credit card company. It functions as a platform that facilitates transactions from two sides: the merchants that pay per payment transaction performed in shops, and the customer that actually performs a payment (Rochet & Tirole, 2003). Accordingly, the credit card company has to stimulate merchants to offer credit card payments as an option to customers, while at the same time the credit card company has to make credit cards attractive for customers so that they will make their payments with such a card at a merchant. Typical in two-sided markets is that there is a certain balance between participation around the platform from different parties and the gained benefits hereof by each individual group of actors (Armstrong, 2006). In the case of a credit card company, for example, the benefits for users increase as more merchants offer the opportunity to pay with a credit card. To address this balance, most previous research stems from economics to determine optimal price and the influence of price competition (Rochet & Tirole, 2003; Armstrong, 2006; Rysman, 2009). As little to no attempt is made by economics to formalize the definition of two-sided markets, this thesis uses the descriptions by Rochet and Tirole (2003) and Rysman (2009) to define two-sided markets as follows:

Two-sided market: A market in which a platform facilitates transactions between at least two distinct groups of actors.

While Gawer (2009a) explicitly includes two-sided markets in the typology, the author argues that not all two-sided markets can be considered as industry platforms, since the vast majority merely facilitates transactions between two groups of actors rather than actually stimulating reuse or thriving innovation. In the context of the software industry, two-sided markets are associated with extension markets such as the Apple AppStore or the plug-in listings for the Internet Explorer web browser (Burkard et al., 2011; Hyrynsalmi, Makila, Jarvi, Seppanen, & Knuutila, 2012). Accordingly, this thesis regards an extension market as a means to facilitate the distribution of complements developed for an industry platform, rather than being a platform on its own. Google Play, for example, harvests the complements that are offered for the industry platform Google Android. This perspective is supported in previous studies conducted by Boudreau (2010), Eisenmann, Parker, and Van Alstyne (2006) and Muller, Kiji, and Martens (2011).

3.1.2 Platform Mechanisms and Characteristics

Industry platforms – in the remainder of this thesis also addressed as platforms – are subject to an array of dynamics, and therefore have been subject of study from different perspectives. This subsection addresses three elaborated aspects of platforms, being network effects, platform architecture and means to captivate value from either opening up or closing platforms. These perspectives originate from the research domains of business strategy and software architecture.

Architecture for modular systems embodies the allocation of functionality to specific components and the facilitation of interaction between different components with interfaces (Ulrich, 1995). In their attempt to identify common aspects in the architecture of industry platforms, Baldwin and Woodard (2009) distinguish between three groups of components that together constitute a platform: (1) a foundation technology, (2) a set of interfaces and (3) complements. In an architectural classification, Iansiti and Levien (2004a) create an abstract representation of complements, by referring to them as implementations, being “proprietary approaches to solving problems” (p. 150). Figure 3.1 represents the architecture of a software platform as described by Baldwin and Woodard (2009) and Iansiti and Levien (2004a).

The foundation technology consists of a number of stable components, that are reused throughout new versions and by complements. The foundation technology is considered

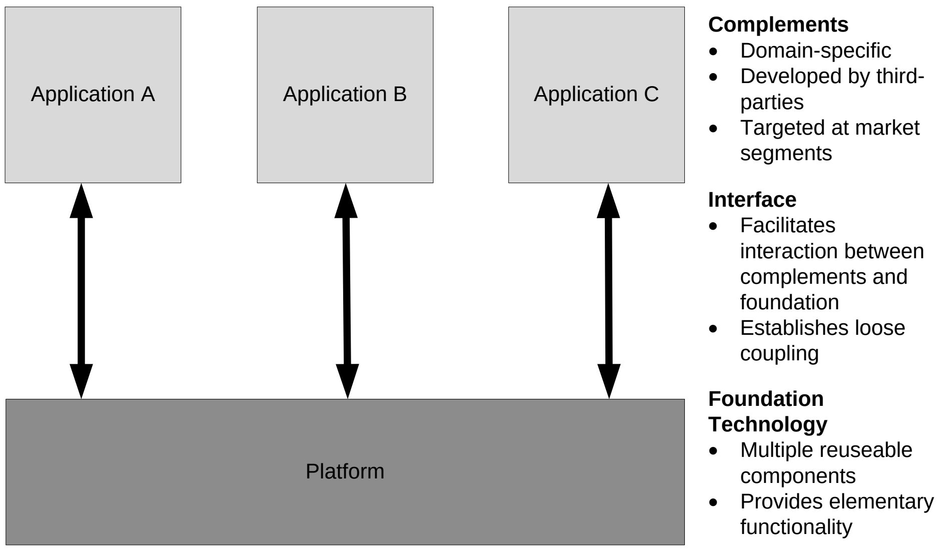

Figure 3.1: The architecture of a platform

stable, though it evolves over time as new versions are released under pressure of innovation (Baldwin & Woodard, 2009). Research with regard to the foundation technology is directed at component reuse, modularization and technological change (Baldwin & Woodard, 2009; Iansiti & Levien, 2004a). Baldwin and Woodard (2009) identify interfaces – in the software industry implemented as APIs – as the most stable component in a platform architecture. Whereas the foundation is prone to innovation, interfaces facilitate the interaction of complements with the foundation technology and therefore have to remain stable, as abrupt changes may harm the interoperability of complements with the foundation technology. In other words, interfaces facilitate loosely coupled integration, meaning that a limited technical effort or lines of code is necessary to integrate with the platform Iansiti and Levien (2004a); Pressman (2004). Iansiti and Levien (2004a) furthermore note that, due to stability of interfaces, platform owners can keep the actual innovation of their platform hidden from complementors, as complementors do not notice any difference in the way in which they integrate and interoperate with the platform.

Table 3.2: Applied platform architecture to Google Android and a shopping mall

Component	**Google Android**	**Shopping mall**
Foundation Technology	Android OS	Building and facilities
Interface	Android API	Retail space
Complements	WhatsApp, Angry Birds	Shop, restaurant

As Baldwin and Woodard (2009) argue that their architectural framework can be applied to both software platforms as physical platforms, Table 3.2 illustrates the application of platform architecture for both a software and physical platform. Table 3.2, respectively illustrates the architecture of the Google Android mobile operating system and a shopping mall.

The value of a platform is dependent on the diversity of complements that are surrounding the platform. Farrell and Saloner (1985) and Cusumano (2010a) refer to this effect as a network effect or network externality surrounding the platform, touching upon the relationship between the number of customers and the value of a product or service. Eisenmann, Parker, and Van Alstyne (2011) argue that all platforms exhibit network effects. Gawer (2009b) and Cusumano (2010a) both conclude that network effects are among the key determinants that make up platform-based competitions in the software industry. Gawer (2009b), for example, notes that the increasing exertion of network effects fosters the competitive position of a platform and its owner, making it harder to dislodge the platform from its power position as the increasing number of products and services provides for an ever increasing cumulative advantage. Cusumano (2010a) then, contributes a proportion of the success of Microsoft Windows to network effects, as the number of applications available for Microsoft Windows attracted an increasing number of users.

Cusumano (2010a), states that a platform ecosystem is composed of: (1) a platform, (2) complements and (3) network effects. Network effects are decomposed in direct and indirect network effects. Direct network effects imply that an increasing number of users of a system leads to increasing value of the system for subsequent users (Farrell & Saloner, 1985). For example, the value of the internet increased as the number of computers that were connected to the internet increased. Direct network effects can be stimulated by means of standardization and compatibility (Farrell & Saloner, 1985), for instance interoperability between personal computer hardware and the Microsoft Windows operating system or the interoperability of complements with the operating system (Cusumano, 2010b). According to Cusumano (2010a), direct network effects are reflected by the growing number of complementary products or services that are made available for a platform.

Indirect network effects, are consumption externalities that result from the purchase of complementary products (Parker & Van Alstyne, 2005). In the context of platforms, indirect network effects can be stimulated through complements, because complements can deliver new functionality that is appealing to an entire new group of customers (Cusumano, 2010a; Zhu & Iansiti, 2012). Consider an ERP platform that becomes attractive for an accountant as a complementor developed a domain-specific extension that incorporates support for critical business processes of an accountancy firm. Furthermore, these consumption externalities can also lead to an increasing interest of other parties, such as service providers or advertisers, that in turn provide increasing variety in the ecosystem. Cusumano (2010b) refers to this effect as a positive feedback loop.

According to West (2003), platform owners face the challenge of balancing appropriability and adoption: that is the balance between reaping benefits from a proprietary platform while these benefits are connected to the adoption of the platform by complementors and end users (e.g. benefits from network effects). Subsequently, researchers have been directing attention at defining platform openness, and providing answers to questions about when to open up a platform and how to achieve this. So far, limited attempts have been made to formalize the concept of platform openness. Instead, researchers focus on factors that determine the 'openness' of a platform. Eisenmann et al. (2009) describe an 'open' platform in accordance with two prerequisites: (1) there are no restrictions on participation in platform development, commercialization or use, and (2) restrictions such as technical requirements or certification are non-discriminatory, meaning that they are uniformly imposed to all participants. In addition, the authors define four different actor groups for which a platform can be made 'open':

- **Demand-side platform users:** end users
- **Supply-side platform users:** complementors
- **Platform providers:** the actors that distribute the platform
- **Platform sponsors:** platform owners

Following from these criteria, iOS is perceived 'closed' as Apple acts as the sole distributor, reserves rights for itself to refuse publishing applications in the AppStore, and access to the foundation technology of iOS is restricted. In contrast, Linux is considered open for all four roles. To formalize 'openness', this thesis builds on the definition of openness by Anvaari and Jansen (2010), who define openness as "the degree to which a platform supplier allows the platform users to interact with the platform, and view or extend its components" (p. 1). As this definition only addresses part of the perspectives on openness introduced by Eisenmann et al. (2009), in this thesis platform openness is defined as follows:

Platform openness: The degree to which a platform owner allows other actors to distribute, extend, use or interact with its platform or one of its associated markets.

As this thesis is formulated around platform ecosystems, platform openness is only addressed from the perspective of complementors. Researchers take different perspectives when evaluating the openness of platforms. Anvaari and Jansen (2010) examine the openness of mobile platforms by looking at the platform architecture. Decomposing a mobile software platform into a layered structure consisting of an application layer, a middleware layer and the kernel layer, the authors examine the extent to which developers can integrate, extend or modify each individual layer. Other authors, such as Boudreau (2010), Gawer and Cusumano (2008), Rysman (2009) and Schilling (2009) regard platform openness in the light of governing complementary markets, involvement of complementors in foundation technology development, the degree of transparency in interface documentation and inter-platform compatibility.

3.1.3 Platform Strategy and Leadership

As researchers articulate, (wannabe) platform leaders are confronted with an array of challenges as they seek to bring and keep their platform to the upfront of their industry. Gawer and Cusumano (2008), however, argue that not every product or service can be turned into a platform. First of all, a product can only be turned into a platform if it performs one or more elementary functions within a system. Second, the potential platform needs to solve a (business) problem for multiple actors or firms. When firms actually can establish a platform, they strive for platform leadership. Gawer and Cusumano (2002), define platform leadership as follows:

Platform leadership: The ability of a firm to drive innovation around a particular platform at the broad industry level.

Among others, platform owners have to deal with technological innovation, platform evolution and the preservation of market leadership (Gawer & Cusumano, 2002). Platform owners are confronted with questions about how to facilitate continuous compatibility with complements, or how to evolve a platform to prevent it from becoming obsolete. To stress the importance of platform strategies, researchers recurrently use the case of IBM and their ever declining position in the personal computer industry as described by Ferguson and Morris (2003). In their successful days, the personal computers of IBM provided for the until recently highest revenue of a single high technology firm. The IBM computers were based on an IBM supply chain platform in which a variety of other companies took part, among which were Microsoft and Intel. As their personal computer was an open platform, IBM became victim of reverse engineering of their own platform by other parties. Furthermore, IBM only possessed one personal computer related standard, BIOS, whereas Microsoft held MS-DOS. As a consequence, IBM slowly lost control over their own platform to their complementors Microsoft and Intel. Gawer (2009a), also illustrates an initially failed initiative of SAP to position themselves as a platform leader. SAP – traditionally a business application vendor – introduced a platform called NetWeaver that should provide for increasing compatibility with non-SAP products, for which complements should be build by their partners. As SAP did not abandon selling its existing products SAP ended up competing with its own ecosystem of complementors, who they initially did want to support in their activities.

As illustrated, different forces are at play in establishing and maintaining an industry-wide strategy. Gawer and Cusumano (2002) define four dimensions – or as they call them 'levers' – that have to be addressed by a successful platform strategy: (1) scope of the firm, (2) product technology, (3) relationships with external complementors and (4) internal organization.

1. **Scope of the firm:** A platform owner has to make decisions about what is done in-house and what is left to complementors. Platform owners have to carefully assess the opportunities that arise to enter complementary markets themselves (Gawer & Henderson, 2007). While attractive to provide for specific functionality that is beneficial for the platform, active participation or aggressive acquisition of complementors that were pivotal in the complementor ecosystem, may force other complementors to depart the ecosystem as they feel the platform owner diminishes the opportunities for complementors (Huang, Caccagnoli, & Forman, 2009; Iansiti & Levien, 2004b; Jansen & Cusumano, 2012). Furthermore, platform owners have mechanisms at their disposal to stimulate innovation by individual complementors, such as disclosing architectural details, sharing expertise, providing venture capitals and engagement into formal partnerships or revenue sharing (Gawer & Cusumano, 2002; Popp & Meyer, 2010).

2. **Product technology:** Platform owners have to make decisions about the architecture of their platform and the surrounding interfaces. Also, platform owners must decide on what information is shared about the platform and, for example, the functioning of its interfaces. Gawer and Cusumano (2002) argue that the choice for a certain foundation technology architecture has an impact on the fate of the platform as a whole. The authors argue that inaccessible architectures may hamper the adoption of the platform by complementors, opposed to a modular architecture with interfaces that reduces entry barriers due to increased transparency and integrativeness (Schilling, 2009). Similarly, sharing technical information about interfaces will help complementors in targeting opportunities around the platform, while on a downside this provides for lowered switching costs – both due to looser

coupling and opportunities to target multiple platforms (Cusumano, 2010a; Pressman, 2004).

3. **Relationships with external complementors:** Platform owners have to make decisions about establishing competitive or collaborative relationships with complementors, and have to manage the network of complementors, for example, by making them collaborate or innovate together, while complementors at the same time engage into competitions (Jansen, Brinkkemper, & Finkelstein, 2009). Additionally, platform owners have to ensure for a degree of consensus, for example, about the standards that are used for interaction between complements and platform in the complementor ecosystem. Gawer and Cusumano (2002) refers to this practice as 'ecological control' and describe it as "balancing consensus and control", whereas Perrons (2009) describe a similar principle as "balancing power and trust".

4. **Internal organization:** Platform owners have to make decisions about how to design and use their organizational structure to avoid internal and external conflicts of interest, departments may be driven by differing incentives. As relationships, in which firms at the same time collaborate and compete are established – better known as co-opetition (Bengtsson & Kock, 2000) – a platform owner may end up with a modular organizational structure, as the platform owner is simultaneously confronted with different dynamics (Patrucco, 2012).

Building on the four pillars of platform leadership, Gawer and Cusumano (2008) identify two generic platform strategies, one for existing platform leaders and one for new entrants that want to build a platform from scratch, respectively referred to as 'tipping' and 'coring'. Important in coring is providing complementors with incentives to become part of the ecosystem, for example, by stimulating the development of complements or by providing complementors a platform (e.g. extension market) to showcase their complements. Part of this is also the provision of opportunities for complementors – the equivalent of niche creation as described by Iansiti and Levien (2004b). Furthermore, Gawer and Cusumano (2008) suggest to opt for a strong protection of intellectual property with regard to platform architecture, and the authors also suggest to increase switching costs for complementors by coupling them tightly with the platform. Tipping involves platform-based competition and focuses on development of appealing features, driving innovation by complementors, subsidizing complementor development or penetration of new markets through complementary products.

3.2 Industries and Markets as Ecosystems

As elaborated in the previous section, industries or ecosystems of complementors play a pivotal role in industry platforms and platform-based competitions. In fact, Baldwin and Woodard (2009) and Iansiti and Levien (2004a) include the complements as one of the core elements in industry platforms. Subsequently, this section provides insight into the research domain of business and software ecosystems, as ecosystems are used as a metaphor to describe and study the interrelatedness of different actors that are part of markets, entire industries or complementor networks (Gawer, 2009b; Jansen, Finkelstein, & Brinkkemper, 2009; Moore, 1993). Initially, the concept of ecosystems has been borrowed from natural systems (Peltoniemi & Vuori, 2004), as Moore (1993) argued that natural ecosystems were self-reinforcing systems that consist of different interdependent species. Translated to the traditional business domain, Moore (1993) stated that firms cannot innovate in a vacuum, rather they depend on other parties to collaborate with them. This collaboration goes beyond industries or traditional supply chains. Moore (1996, p.26) defines a business ecosystem as follows:

Business ecosystem: An economic community supported by a foundation of interacting organizations and individuals.

In a business ecosystem, the economic community develops goods and services that are of added-value to customers, who are themselves part of the ecosystem (Moore, 1996). The business ecosystem of a car manufacturer consists of all its suppliers, partners, distributors and customers. Over time, the members in the ecosystem co-evolve their capabilities which is coordinated by one or more central players. Moore (1993) regarded business ecosystems as a new metaphor to explain complex affairs in business environments, such as the fall of IBM in the personal computer market that has been discussed in the previous section.

While the interest for business ecosystems primarily stemmed from technological domains, researchers (Jansen & Cusumano, 2012; Messerschmitt & Szyperski, 2005) argued that business ecosystems did not put enough emphasis on the specific characteristics and dynamics at play in the software industry to function as a valuable metaphor. Contrary to physical products, software products can be reproduced at a fraction of its original costs, therefore the main boundaries confronted by software companies are of conceptual or economical nature (Jansen & Cusumano, 2012; Messerschmitt & Szyperski, 2005; Xu & Brinkkemper, 2007). First introduced by Messerschmitt and Szyperski (2005), software ecosystems address the identified deficiency by putting emphasis on the unique characteristics of the software industry. Software ecosystems contain all software related organizations and individuals that are part of the overall business ecosystem (Jansen & Cusumano, 2012). In this thesis, a software ecosystem is defined as follows (Jansen, Brinkkemper, & Finkelstein, 2009, p. 2):

Software ecosystem: A set of actors functioning as a unit and interacting with a shared market for software and services, together with the relationships among them.

The remainder of this section defines the main concepts of ecosystems. Among others, this section discusses the foundation of a software ecosystem, its two core components being actors and relationships, and the management of ecosystems alongside its parallels with platform strategies. As software ecosystems are considered to be a subset of business ecosystems (Jansen & Cusumano, 2012), literature from both domains is used interchangeably.

3.2.1 Ecosystem Perspectives and Scope Levels

After the preliminary introduction of software ecosystems by Messerschmitt and Szyperski (2005), Jansen, Finkelstein, and Brinkkemper (2009) and Jansen, Brinkkemper, and Finkelstein (2009) did propose mechanisms to aid in structuring research on software ecosystems, since the authors found that research on software ecosystems differed in terms of scope. Accordingly, Jansen, Finkelstein, and Brinkkemper (2009) introduce three scope levels for software ecosystems: (1) software ecosystems level, (2) software supply network level and (3) organizational level, which are represented in Figure 3.2 (Jansen, Brinkkemper, & Finkelstein, 2009, p.2).

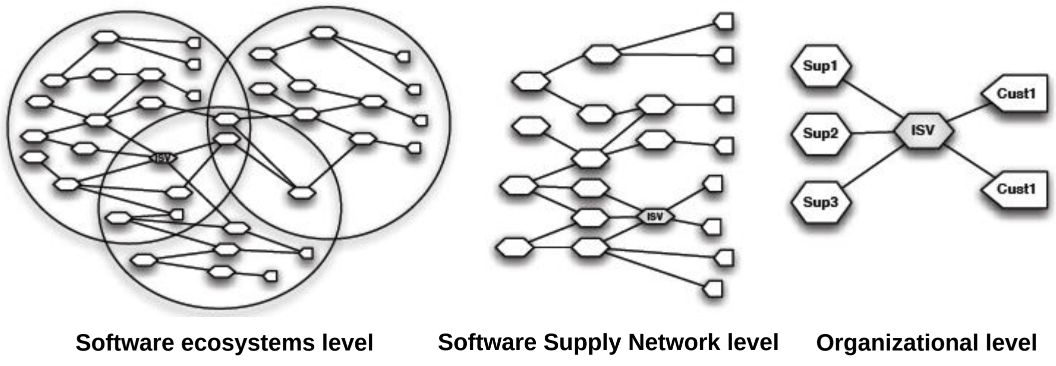

FIGURE 3.2: Three scope levels in software ecosystem research

The organizational scope level, encompasses the perspective on relationships for an individual organization. The scope level is involved with decisions about how a single firm can create value for, or leverage the ecosystem (Jansen, Brinkkemper, & Finkelstein, 2009). Organizations are facing decision-making about delivery models, product planning and the facilitation of portability and platform extensions (Jansen, Finkelstein, & Brinkkemper, 2009). The software ecosystem level, regards entire ecosystems and the relationships among them. This scope level, for example, could be used to address the strategic decisions of platform owners to be compatible or incompatible with competing platforms, as discussed by business strategists and economics such as Rysman (2009). The software supply network level, regards a software ecosystem as a series of connected software supply networks. Decisions at this scope level are directed at monitoring both indirect and direct buyer-supplier relationships (Jansen, Brinkkemper, & Finkelstein, 2009). In this context, a software supply network is defined as follows (Jansen et al., 2007, p. 1):

Software Supply Network: A series of linked software, hardware and service organizations that cooperate to satisfy market demands.

Apart from the definition of scope levels, researchers have also been involved with the definition of ecosystem boundaries. As Jansen, Finkelstein, and Brinkkemper (2009) articulate, relationships in ecosystems are frequently 'underpinned' by a common technological platform or market, and operate through the exchange of information, resources and artifacts. In the case of a platform ecosystem, for example, the foundation technology and its interfaces are considered as the 'underpinnings' of the ecosystem. The platform owner then together with its customers, and complementors together with their customers, constitute the platform ecosystem. Additionally, Jansen, Brinkkemper, and Finkelstein (2009) name three other types of ecosystems based on their mediating platform:

- **Market ecosystem:** enterprise resource planning ecosystem, game ecosystem
- **Technology ecosystem:** .NET ecosystem, IPv6 ecosystem
- **Firm ecosystem:** Google ecosystem, Microsoft ecosystem

In a different typology, Bosch (2009) puts central emphasis on the importance of platforms in ecosystem theory. Based on the typology, one can distinguish between desktop platforms, web platforms and mobile platforms, that are operating systems, applications or end-user programming.

3.2.2 Actors and Relationships

Actors play a pivotal role in ecosystems, as they engage in relationships and provide for interaction, exchange and innovation with and among each other. Actors can shape (Hagel III, Brown, & Davidson, 2008; Iansiti & Levien, 2004a) and orchestrate (Jansen, Finkelstein, & Brinkkemper, 2009; Jansen, Brinkkemper, & Finkelstein, 2009), but can also be just participating within one or more ecosystems, for example, by developing complements for a platform (Gawer & Cusumano, 2002). In previous work (Van Angeren, Blijleven, et al., 2013, p. 6), an actor has been defined as follows:

Actor: An entity that shapes, orchestrates, participates in, or contributes to a software ecosystem depending on its role, type and purpose.

Attempts have been made to classify actor groups that share similar characteristics. Hagel III et al. (2008) distinguish between shapers (e.g. the actors that form and develop the ecosystem) and participants that benefit from the ecosystem effort of the shaper in creating business value by joining an existing ecosystem. Earlier, Iansiti and Levien (2004a), Iyer et al. (2006) and Jansen, Brinkkemper, and Finkelstein (2009) employed a comparable classification that incorporates the following roles:

- **Keystone:** An actor that provides the foundation for (part of) the ecosystem with its standards, technology or platform.
- **Dominator:** An actor that progressively assimilates (e.g. through mergers and acquisitions) and eliminates other actors within the ecosystem.
- **Niche player:** An actor that requires the technology, standard or platform of the keystone to create business value.

Apart from an abstract classification in three distinct roles, actors can be classified from a variety of perspectives. Popp and Meyer (2010) for example, classify actors based on their purpose and actions within the ecosystem, influenced by their business model archetype and the products or services they offer. Examples of roles that the authors distinguish are inventors, contractors and distributors. Previous attempts on classifying actors based on their types also exist. Kittlaus and Clough (2009) identified the following actor types: (independent) software vendor, value-added reseller, value-added distributor, original equipment manufacturer, system integrator and a technological alliance.

From a networked perspective, Iyer et al. (2006) distinguish between hubs, bridges and brokers, referring to their role in stimulating diffusion throughout the network. In their classification, Iyer et al. (2006) put emphasis on relationships between actors, as the number of relationships determines how influential a keystone is. Similarly, a broker creates a connection between two sets of firms that otherwise would stay disconnected (Iyer et al., 2006). Relationships have been a recurring research subject in business studies. From an ecosystem perspective, previous work (Van Angeren, Blijleven, et al., 2013, p. 5) resulted in the following definition of a relationship:

Relationship: A connection between two or more actors, characterized by the flow of products, services, money, and intellectual property.

Interfirm relationships come in numerous forms, examples of which are alliances (Iyer et al., 2006; Lavie, 2007; Rosenkopf & Schilling, 2007), distribution partnerships (Jansen, Brinkkemper, & Finkelstein, 2009; Kittlaus & Clough, 2009; Popp & Meyer, 2010) partnership model participation (Van Angeren et al., 2011; Van Angeren, Kabbedijk, et al., 2013), shared research and development (Santamaria & Surroca, 2011), and technological partnerships (Basole, 2009). The rationale behind relationships initiation also differ. Technological partnerships may be initiated to establish chains of interoperability in an ecosystem (Iyer et al., 2006), being mutually compatible software products or complementarities. Other motives include necessity (Basole, 2009) extension of the value network (Varis, Kuivalainen, & Saarenketo, 2005) and the acquisition of complementary resources or capabilities (Dyer & Singh, 1998).

Inspired by social sciences, business researchers have been involved with characterizing relationships. Håkansson and Snehota (1995), for instance, elaborate on the presence of asymmetry in relationships, caused by differences in company sizes or resources available to organizations, that result in a disturbed balance between organizations, providing changing bargaining positions. Another parallel between social networks and ecosystems is that both domains consider relationships as the main characterizing factor of a network (Hanneman & Riddle, 2005; Jansen, Brinkkemper, & Finkelstein, 2009).

3.2.3 Ecosystem Management and Orchestration

In their four levers of platform leadership that were described previously, Gawer and Cusumano (2002) put emphasis on the management of external relationships with complementors. Gawer and Cusumano (2002) stress the need for collaboration, stimulation of innovation and niche preservation in the ecosystem, as the strength and diversity in the complementor ecosystem is a determinant for success in platform-based competition. Whereas Gawer and Cusumano (2002) include ecosystem management as one of the factors in achieving platform leadership, Iansiti and Levien (2004a) place ecosystem management central in their keystone strategy. Iansiti and Levien (2004a) argue that individual species thrive when the ecosystem they are part of is successful. A keystone thus should be after preservation of prosperity of the ecosystem. Prosperity herein is constituted of the following three determinants (Iansiti & Levien, 2004a, 2004b):

1. **Robustness:** The capability of an ecosystem to face and survive disruptions.
2. **Productivity:** The efficiency with which an ecosystem converts inputs into outputs.
3. **Niche Creation:** The capacity to create meaningful diversity and thereby new opportunities.

According to the operationalization of ecosystem health measurement by Den Hartigh et al. (2006), robustness can among others be measured by looking at the survival or bankruptcy rates of niche players in the ecosystem. Productivity can be measured by means of economic productivity indicators and by looking at how the ecosystem leverages innovation, and niche creation can be measured by performing species analysis (e.g. classifying actors that constitute the ecosystem). In evaluating their ecosystem health measurement operationalization, Den Hartigh et al. (2006) perform case studies that evaluate the individual and overall partner health of partnership models employed by multinationals in the software industry. In this evaluation, the authors touch upon another aspect of ecosystem management: actor and relationship management. Den Hartigh et al. (2006), for example, argue that relationships among different actors of the same species may be regarded as an indicator of intensifying competition in part of the ecosystem. Similarly, a large number of clusters in an ecosystem are by the authors considered as an indicator of increasing collaboration and innovation in the ecosystem.

In their shaping strategy, Hagel III et al. (2008) emphasize the need to build or redesign ecosystems. The authors oppose their shaping strategy to an adoption strategy, which merely entails reacting to technological change and defending existing markets. A shaping strategy consists of three elements. First of all, a shaper needs a shaper view with which it can draw (economic) perspectives for potential participants, in such as a way that opportunities are provided for new entrants. Second, a shaping strategy needs a shaping platform, a set of standards and practices that helps to support and organize efforts in the ecosystem. Hagel III et al. (2008), distinguish between two forms of platform leverage: technological leverage, being the foundation technology and interfaces of an industry platform, and interaction leverage referring to two-sided markets. Last, a shaper needs shaping acts with which it can stimulate adoption of the ecosystem, ranging from decreasing entry barriers to signaling commitment to the platform.

3.3 Visualization and Analysis of Ecosystems

The first two sections of this chapter reflected the commonalities in platform-centric and ecosystem-centric thinking. Both perspectives put strong emphasis on network value, co-opetition, reaping benefits from network effects, and distinct roles of complementors and platform owners. Subsequently, this section discusses the analysis that takes place on the intersection of platforms and ecosystems – the analysis of platform ecosystems. In a retrospective survey of previously applied platform ecosystem analysis, Baldwin and Woodard (2009) identify three dominant approaches, being (1) layer maps, (2) network visualization and (3) design structure matrices. The remainder of this section addresses and illustrates the application of each of these mechanisms, alongside a discussion of their strengths and shortcomings.

3.3.1 Layer Maps

Software stacks are a popular approach to documenting the architecture of a software product or platform (Gao & Iyer, 2006). In the vertically integrated perspective of a stack view, a platform is decomposed into layers, each fulfilling part of the functionality of a system. Typically, each advancing layer can utilize the functionality of a previous layer. Baldwin and Woodard (2009) and Gao and Iyer (2006) articulate that a stack perspective can apart from platforms also be used to structure an entire industry. Each part of the stack then contains multiple parties that develop substitutes. Gao and Iyer (2006), for example, divide the software industry from lowest to highest layer into: hardware, system software, middleware services, application software and services, while Fransman (2002) modeled the telecommunication industry by means of the same approach. Layer maps are constituted of vertical silos (e.g. software stacks) that describe the structure of an industry or platform (Baldwin & Woodard, 2009). Each individual layer then, categorizes the actors that compete against each other with substitutes. To illustrate the principle of layer maps, Figure 3.3 shows the layer map of the computer industry (Baldwin & Woodard, 2009, p. 18).

Baldwin and Woodard (2009) consider layer maps an effective way to provide insight into co-opetition within an ecosystem, as layer maps aid in the identification of actors that provide similar complements. Furthermore, layer maps can be used in longitudinal

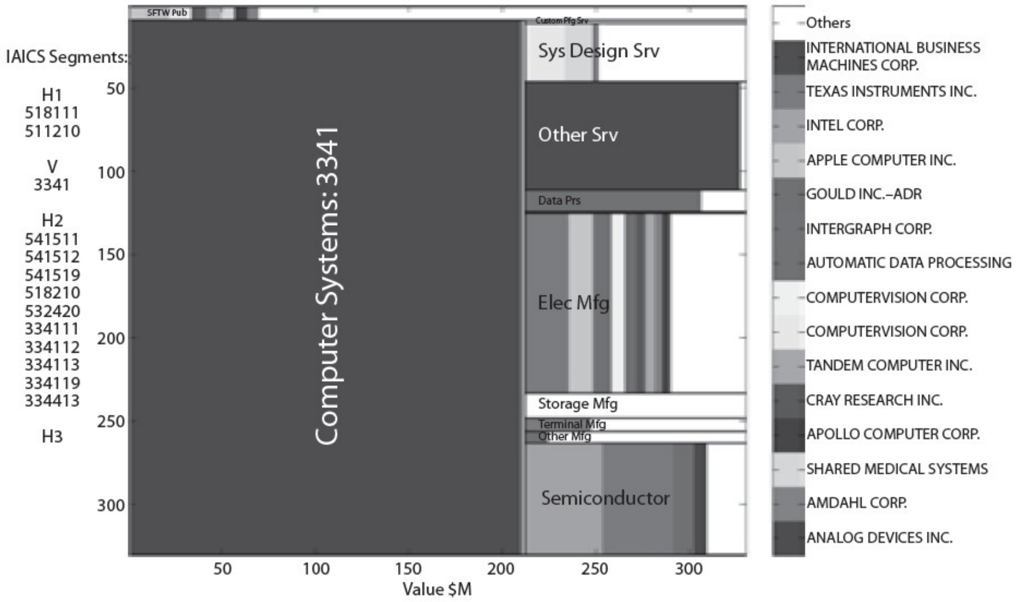

FIGURE 3.3: Layer map of the computer platform

studies of platforms, as layer maps can reveal changes in the ecosystem structure or architecture over time. Despite its advantages, Baldwin and Woodard (2009) note that the construction of layer maps leaves room for debate, as it is found difficult to categorize complements within set boundaries. This is prevalent in ecosystems that form around industry platforms such as (mobile) operating systems, enterprise applications or web browsers. The apps available in the extension market usually merely consist of several thousand lines of code (Wasserman, 2010) and solve one end-user need. Functional classification thus results in proliferation of layers due to the high variety of applications. Accordingly, layer maps are not applicable for this research project.

3.3.2 Network Graphs

Network graphs find their origins in social network analysis and are used to visualize complex interrelated systems (Hanneman & Riddle, 2005; Scott, 2000). Networks are visualized as a function of nodes (e.g. actors, businesses) and edges that represent the relationships among them. Baldwin and Woodard (2009) label network graphs as the predominantly adopted means to visualize platform ecosystems. A centralized platform ecosystem – that is constituted of one central actor surrounded by a number of complementors – is visualized as one central node surrounded by a 'cloud' of complementor nodes. Figure 3.4 shows the IBM example of such a centralized ecosystem (Iyer et al., 2006, p. 21). According to Baldwin and Woodard (2009), such a network graph can provide insight into the existence of one or more central actors, the size of the 'cloud' of complementors and the development of the ecosystem over time by comparing network graphs of the same ecosystems constructed at different points in time. Basole (2009) regards network graphs as a semantically rich way to visualize complex phenomena such as inter-firm relationships. Opposed to its effectiveness, Baldwin and Woodard (2009) and Iyer et al. (2006) mention the increasing complexity (e.g. cluttering), and the incapability of network graphs to denote directional dependencies as shortcomings of network graphs.

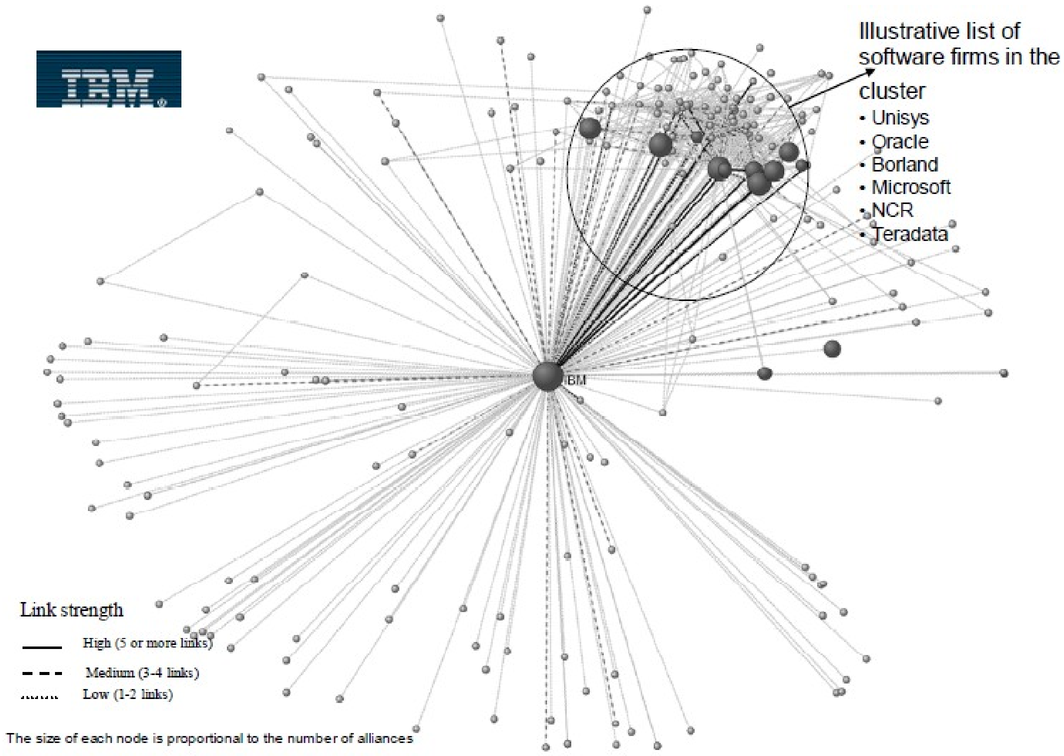

FIGURE 3.4: Network graph of the IBM ecosystem

Apart from visual observation, network graphs are subjected to network analyses (Burt & Minor, 1983; Hanneman & Riddle, 2005; Scott, 2000). Network analysis is comprised of a set of qualitative measures that describe the structure of a network (Scott, 2000), inspired by the degree of centrality of an actor as defined by Freeman (1979). Researchers on software ecosystems embraced network graphs and network analysis to study and analyze software ecosystems. Jansen, Finkelstein, and Brinkkemper (2009), for example, call for quantifiable characterization of ecosystems based on network metrics. Case studies have led to the visualization of ecosystems such as Ruby on Rails (Kabbedijk & Jansen, 2011), the free open source community that contributes to projects hosted on SourceForge (Madey et al., 2004), the mobile phone manufacturer ecosystem (Basole, 2009), the mobile payment ecosystem (Iyer, 2012), the radio frequency ID ecosystem (Quaadgras, 2005) and IBM, Microsoft and SAP (Iyer et al., 2006). While previous visualization studies apply network analysis, these analyses are based on a small set of network metrics. As the studies put emphasis on other descriptives of the ecosystem, rather than the overall network structure.

Network graphs are usually constructed by means of adjacency matrices (Hanneman & Riddle, 2005). An adjacency matrix is made up of as many rows and columns as there are actors in a network. In every row or column, the presence or absence of a relationship is indicated by means of numeric value. Not only does this preserve systematic input of relational data, it also allows comparison of networks by means of matrix algebra and advanced statistics. Network graphs can thus be compared at three different levels: (1) by means of visual inspection of network graphs, (2) through comparison of network metrics and (3) with matrix algebra and statistical analysis. The combination of these three insights provides diverse inputs, while at the same time means to find statistical evidence for these inputs. Accordingly, this approach is applied in the remainder of this thesis.

3.3.3 Design Structure Matrices

Complex software systems – such as industry platforms – consist of components that depend on each other to generate output (Jansen et al., 2008; Sanchez & Mahoney, 1997). If a platform, for example, facilitates extensions through interfaces, complements are obliged to depend on these interfaces, that as a consequence will remain stable (Baldwin & Woodard, 2009). Originating from the domain of software engineering, design structure matrices are designed to visualize dependencies between different components of a system, inspired by traditional two by two matrices (Baldwin & Woodard, 2009; Browning, 2001). In a design structure matrix, the contents of both axes are equal, being the names or labels of a component. For each component, the assessor determines what other components it depends on, by filling the row of this respective column. Figure 3.5 illustrates an example design structure matrix (LaMartina, Cai, MacCormack, & Rusnak, 2008, p. 6). Reading down a column reveals input dependencies whereas reading along a row indicates output dependencies.

Baldwin and Woodard (2009) and Browning (2001) regard design structure matrices as an effective way to visualize complex systems that are composed of thousands of components, as cluttering is non existing in a design structure matrix. Furthermore, Baldwin and Woodard (2009) find design structure matrices effective to identify dependencies among complements and to study the evolvability of a platform. In Figure 3.5, for instance, the block on the bottom-left of the figure represents licensed-in code. The code was deliberately separated from the kernel of the platform, and is thus replaceable (LaMartina et al., 2008). Opposed to its advantages, the creation of design structure

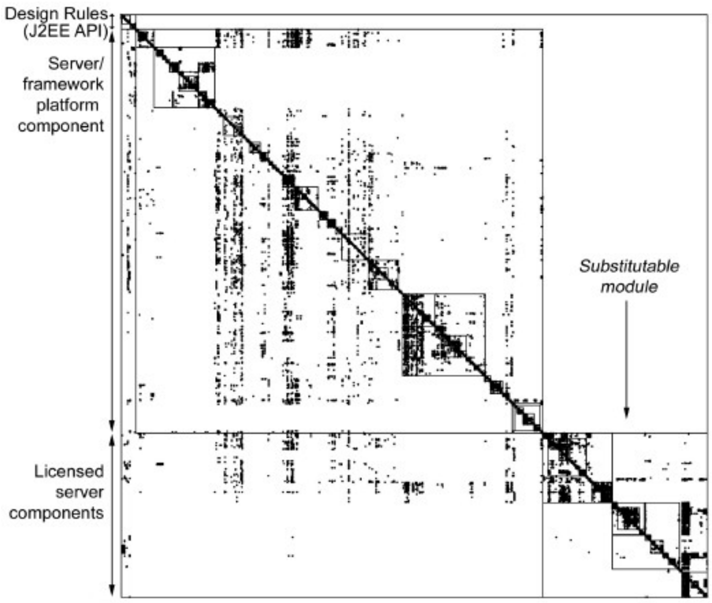

FIGURE 3.5: Design structure matrix of a modular platform

matrices requires an extent of prior knowledge about the architecture of the platform (Baldwin & Woodard, 2009). As a consequence, design structure matrices are not found suitable for executing platform studies from an external perspective – such as the analysis that is performed in this thesis.

Chapter 4

The Operationalization of Network Analysis for Ecosystems

Social network theories are in many ways similar to the ecosystem metaphor that is used as a referent to study the interconnected business domains within or outside of the software industry. Similar to ecosystems, social networks are defined as a composition of actors (e.g. social entities such as people, employees or organizations) and the social relationships among them (Garton, Haythornthwalte, & Wellman, 1997). While sociologists adopted network graphs – referring to them as 'sociograms' – as a means to visualize social networks such as communities, businesses or neighborhoods in a systematical manner, graphs originate from computer science and mathematics (Hanneman & Riddle, 2005). From the 1960s onward, computer scientists and mathematicians have been using what they call 'graph theory' to study mathematical structures, that consist of objects that are pair wisely connected (Bollobas, 1998; Golumbic, 2004). From the perspective of mathematics and computer science, a graph consists of adjacency vertices, being nodes that are directly connected by edges or links (Golumbic, 2004). The total network then, is constituted of all adjacency pairs that are part of a given population or structure (Golumbic, 2004). Throughout the years, mathematicians and computer scientists have been building an extensive body of knowledge on graph theory, illustrated by the large number of books available on the topic.

While network graphs alone proved valuable in representing and visualizing networks, the visualizations themselves were not found suitable for in-depth and automated analysis. To address this deficiency, Festiger (1949) proposed to adopt matrix algebra and consequently adjacency matrices from mathematics. The systematic of an adjacency matrix is comparable to that of a design structure matrix: for every actor in a network, an assessor identifies with which actors it has a direct relationship, and qualifies the (absence of) a relationship with a value. If the measurement is binary, the assessor assigns‘0’ if a relationship is absent, whereas ‘1’ is used to denote the presence of a relationship.

Adjacency matrices and their visualizations are subjected to formal network analysis methods that also reside from the domain of mathematics (Hanneman & Riddle, 2005; Scott, 2000). Formal network analysis methods can be used to uncover characteristics of a network, that include its cohesion, degree, centrality or to determine the influential position of individual actors. Herein, a distinction is made between *network level metrics* that describe the characteristics of the ecosystem as a whole, and *node level metrics* that determine the characteristics of individual actors in the network. So far, existing ecosystem visualization studies have predominantly been involved with the quantification of network centralization and density.

As methods for network analysis require knowledge about a specific vocabulary, the remainder of this section defines and illustrates the network metrics that will be used to perform network analysis in the empirical part of this research. Metrics are formalized by means of mathematics to stimulate replication. To focus on the applicability of network metrics in the context of platform ecosystems, parallels will be drawn based on findings obtained from previously conducted network studies. A complete overview of the network metrics described in this chapter can be found in Table 4.1.

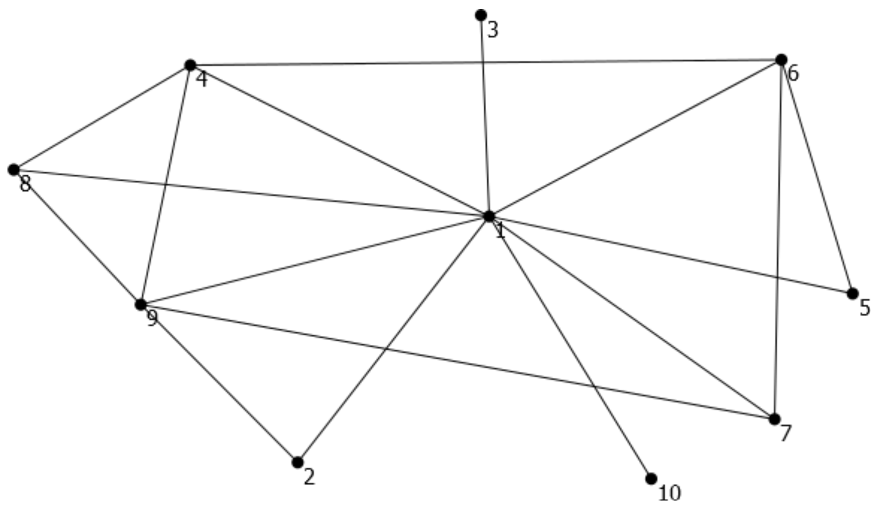

FIGURE 4.1: Network graph of the example platform ecosystem

To provide for illustration in the remainder of this section, Figure 4.1 shows a sample network. This sample network symbolizes a small platform ecosystem, in which actor 1 is the platform owner. The other nine actors then, offer complements to the software platform and are found in the periphery of the graph. Relationships in the platform ecosystem are undirected, meaning that if actor 1 has a relationship with actor 2, actor

2 is also supposed to have a relationship with actor 1. Furthermore, the graph is binary and thus only considers the existence or absence of a relationship.

Table 4.1: Definition of network metrics

Category	Metric	Definition
Node level	Degree centrality	The ratio of the number of adjacent relationships an actor has to the number of adjacent relationships that are theoretically possible.
	Eigenvector centrality	The position of an actor in a network determined by the networked position of its adjacencies.
	Bonacich power	The influential position of an actor in a network determined by the centrality of its adjacencies.
	Clustering coefficient	The ratio of the number of adjacencies of an actor that is also connected to one another to the total number of adjacencies an actor has.
Network level	Size	The number of actors in an ecosystem.
	Network density	The ratio of the number of relationships that are actually present in the ecosystem to the number of relationships that are theoretically possible.
	Centralization	The ratio of the structural position of the platform owner to the centrality of all other actors within the ecosystem.
	Modularity	The degree to which a network can be divided into sub groups that are densely connected to each other, while sparsely connected to actors belonging to other sub groups.
	Clustering coefficient	The degree to which the adjacencies of actors in the ecosystem are also connected to each other.

4.1 Size and Density

The size and cohesion of a network are among the easy to observe, yet critical characteristics of social networks or ecosystems. Social theorists classify networks based on their size, referring to larger networks as being complex and difficult to govern (Hanneman & Riddle, 2005). In describing and measuring the success of ecosystem strategies, ecosystem researchers use the number of actors in an ecosystem as a performance indicator

(Gawer & Cusumano, 2002; Den Hartigh et al., 2006; Jansen, Brinkkemper, Souer, & Luinenburg, 2012; Van Angeren et al., 2011). Herein, an increasing number of actors indicates a growing number of complements for a platform or increasing product adoption. Network size can thus be used to compare adoption levels for competing platforms, and as such provides an easily observable indicator in subsequent analyses. The size of an ecosystem is determined by the number of actors it is constituted of, and accordingly is defined as follows:

Size (S): The number of actors in an ecosystem.

In the example ecosystem shown in Figure 4.1, the network size is equal to ten. To formalize this finding, let G be the ecosystem that is represented by a graph and let V be the number of actors in G. The size of the graph G then, is equal to $|V|$, as shown in Equation 4.1.

$$S = |V| \tag{4.1}$$

Further observation of Figure 4.1 reveals that, although the network appears to be well connected (e.g. all actors are reachable through other actors), not all possible relationships are present. The absence of a considerable number of ties means that the network density is relatively low (Hanneman & Riddle, 2005; Scott, 2000). Granovetter (1976, p. 3) defines network density as follows:

Network density (Δ): The ratio of the number of relationships that are actually present in the ecosystem to the number of relationships that are theoretically possible.

The network density thus, is dependent on the size of the network as it determines the number of relationships that theoretically can be established. The maximum number of ties an actor can establish is equal to *V-1*, but has to be divided by two because of the symmetric dataset. Network density can reach values from 0 (i.e. empty graph) to 1 (i.e. fully connected graph). If E is the number of relationships among actors in a platform ecosystem, then Equation 4.2 shows the computation of network density (Δ).

$$\Delta = \frac{2E}{V(V-1)} \tag{4.2}$$

According to Basole (2009), network density represents the degree of interaction that takes place within an ecosystem, supporting the notion that densely connected ecosystems are more likely to thrive innovation and exchange around a platform. Iansiti and

Levien (2004b) and Den Hartigh et al. (2006) argue that the robustness of an ecosystem is partly determined by the degree of cohesion of the ecosystem. In their platform strategies, Gawer and Cusumano (2008) also emphasize the need to tightly embed complementors in the ecosystem, as well embedded actors are less prone to forces that will make them depart the ecosystem. Throughout subsequent analyses in this thesis, network density is used as an indicator for the degree of collaboration among complementors in platform ecosystems.

4.2 Centrality and Power

Asymmetry between actors can lead to shifting power positions in business networks (Håkansson & Snehota, 1995). While Håkansson and Snehota (1995) in their description of symmetry explicitly refer to forces such as available resources or company size, the concept of asymmetry can also be applied to networks. Sociologists and ecosystem theorists emphasize that power is inherently relational (Bonacich, 1987; Iansiti & Levien, 2004b; Hanneman & Riddle, 2005; Iyer et al., 2006; Venkatraman, Lee, & Iyer, 2008; Basole, 2009; Fricker, 2009). The structural position of an actor within the network imposes constraints and opportunities for an actor (Hanneman & Riddle, 2005). Actors that have a large number of relationships with other actors occupy central positions in the network, and thus find themselves in a favorable position. A keystone, for example, is considered to be a central hub in the ecosystem, referring to its richly networked position (Iansiti & Levien, 2004a, 2004b). Power in this sense is measured by the number of relationships an actor has, compared to the maximum number of relationships that it could theoretically have. In graph theory, this metric is known as degree or point centrality (Freeman, 1979; Scott, 2000), and is defined as follows:

Degree centrality (C_i): The ratio of the number of adjacent relationships an actor has compared to the number of adjacent relationships that are theoretically possible.

As the degree centrality of actors is a micro level measurement, it is a descriptive for individual actors and their structural position in the network. The centrality measurement, however, can also be applied to entire ecosystems, and is referred to as centralization (Freeman, 1979) or global centrality (Scott, 2000). In a star-shaped network – a network with one central actor surrounded by complementors that do not initiate relationships among each other – the centrality is equal to 1. Hence, if one complementor refuses to release its complements through the extension market of the platform owner, the complementor will become isolated from participation in the ecosystem as a whole. Centralization, as introduced by Freeman (1979) compares the network that is subject of study to the star-shaped network, and uses a percentage-scale to indicate its (dis)similarity. A high percentage herein indicates the presence of central actors, whereas a low score indicates equality among actors in the ecosystem (Scott, 2000; Hanneman & Riddle, 2005).

Centralization (C): The ratio of how central the most central actor is in relation to how central all the other actors are.

The centralization score of a network is dependent on the degree centrality score of the actors that populate the network. The more actors engage into relationships with each other, the more decentralized the network becomes. This degree centrality can be formalized in accordance with its definition. Let a_{ij} denote an adjacency pair in adjacency matrix A that structures a graph G, that has value 1 if actor V_i and V_j have a relationship and 0 otherwise. The sum of all these adjacency pairs equals the degree of an actor. If $E_i\text{-}1$ is the maximum number of relationships an actor can theoretically have, the degree centrality of an actor can be computed as follows (Freeman, 1979):

$$C_i = \frac{\sum_{i=1}^{n} a_{ij}}{E_i - 1} \tag{4.3}$$

In platform ecosystems, the degree centrality of the platform owner will always be equal to 1, as the platform owner is considered to have a relationship with every other actor. Centralization, compares the degree centrality of this platform owner (C_{max}) to the centrality of other actors in the network (C_i). Subsequently, centralization is the ratio between the observed differences in degree centrality to its maximum value (Freeman, 1979). The resulting equation is shown in Equation 4.4 and equals 0 in case all actors have the same number of connections.

$$C = \frac{\sum_{i=1}^{n} C_{max} - C_i}{max\left(\sum_{i=1}^{n} C_{max} - C_i\right)} \tag{4.4}$$

Apart from the ego-centric perspectives of degree centrality and centralization, Freeman (1979) also introduced betweenness and closeness centrality that regard the structural position of an actor from the context of the bigger network. Closeness centrality measures the 'nearness' of an actor to all other actors in the same network by means of direct and indirect ties (Freeman, 1979; Hanneman & Riddle, 2005). An actor with a high closeness centrality is considered powerful as it has easy access to complementary knowledge and resources possessed by other actors. Betweenness centrality regards the centrality of an actor by how central it is in connecting other actors with each other (Freeman, 1979). Since the research described in this thesis addresses direct collaboration among complementors, both betweenness and closeness centrality were not found suitable metrics for subsequent analyses.

Contrary to Freeman (1979) – who regards every relationship as equally important – Bonacich (1987) argues that not each adjacency is equally important. According to Bonacich (1987), an actor has increased power when it acts as a hub to less densely connected parts in the network. In Figure 4.1, for example, actor 6 can be an important hub in actively engaging actor 5 and 7 in the diffusion of innovation within the ecosystem. Simultaneously, Bonacich (1987) states that an actor that is located in a dense part of the network is central, compared to an actor that is located in the periphery of the ecosystem with a low density, even if it has a high degree.

Eigenvector centrality (Bonacich, 2007), determines the centrality of actors based on their farness or closeness to other actors in the network, compared to degree centrality that only looks at the immediate neighborhood of an actor (i.e. ego network). As such, eigenvector centrality could be considered as a comparable indicator to betweenness or closeness centrality. However, eigenvector centrality puts stronger emphasis on the global network topology. Consequently, the centrality of an actor is determined by its own degree and the centrality of its adjacencies. The computation of eigenvector centrality is an iterative process that is based on the mathematical approach of factor analysis (Hanneman & Riddle, 2005), an elaboration of which is beyond the scope of this thesis. The eigenvector centrality of the entire ecosystem, is the sum of the eigenvector of all actors.

Eigenvector centrality: The position of an actor in a network determined by the networked position of its adjacencies.

To formalize the iterative computation of eigenvector centrality, let the eigenvector centrality of actor i be denoted x_i. Meanwhile, λ is a constant with a value that is ≥ 0. If the eigenvector centrality of i is proportional to the eigenvector centrality of its adjacencies, the x_i can be computed as shown in Equation 4.5. Subsequently, the average eigenvector centrality for the entire ecosystem is shown in Equation 4.6.

$$x_i = \frac{1}{\lambda} \sum_{j=1}^{n} a_{ij} x_j, \tag{4.5}$$

$$x = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.6}$$

The power index introduced by Bonacich (1987) is based on eigenvector centrality. In addition to the original formula, Bonacich (1987) introduced a new parameter β, that can have a value between -0,5 and 0,5. A low β favors actors that are hubs to the periphery, as relationships to well connected adjacencies are weighted negatively. A high β on the contrary, favors relationships with well connected actors, providing for similar indications as eigenvector centrality. When β approaches zero, the Bonacich power index equals degree centrality. The power index defined by Bonacich (1987) can be used to evaluate the bargaining position of actors in networks in which reputation, weighted relationships or co-opetition are principal determinants, and therefore provides a valuable indicator for subsequent analyses.

Bonacich power(C(α, β)): The influential position of an actor in a network determined by the centrality of its adjacencies.

Apart from a predefined β, the computation of the Bonacich power index also depends on the predefinition of a scaling vector α to normalize the indices. Let A be the adjacency matrix that consists of a_{ik}, and let I be the identity matrix of A (i.e. the equivalent of A in which the diagonal has value 1). If 1 is a matrix that solely contains the value 1, then the Bonacich power index is computed as shown in Equation 6.1 (Bonacich, 1987).

$$C(\alpha, \beta) = \alpha(I - \beta A)^1 A1 \tag{4.7}$$

Centrality – while still scarcely illustrated in empirical studies – is proposed as a pivotal measure in ecosystem analysis. Den Hartigh et al. (2006), for example, operationalize the degree centrality by Freeman (1979) to determine the persistence of an actor within the overall ecosystem. Herein, central actors are labeled as healthy, compared to their peripheral siblings. Quaadgras (2005), argues that firms should take their centrality in ecosystems into account as part of their alliance strategy, so that future alliances can benefit the network position of a firm. Basole (2009) uses the centralization and network density of parts of the network to qualify a segment as being either current or emerging. The author finds that emerging segments have a low centrality, as actors are relatively new in the ecosystem, and thus share limited business relationships in the ecosystem.

4.3 Ecosystem Clusters

Pivotal in network analysis is the identification of sub groups or clusters in the ecosystem (Scott, 2000; Hanneman & Riddle, 2005). A sub group is a set of actors that is more closely tied to one another than to actors that are not part of the group (Hanneman & Riddle, 2005). According to social scientists, decomposing the entire network into sub groups aids in understanding the solidarity and democracy in the ecosystem (Hanneman & Riddle, 2005). A network that consists of two sub groups, for example, may be compared to a landscape in which two political parties with opposite ideals contrast. Also, an actor that is central in a large cluster is an influential actor in the overall network. Thus, the position of an actor in a sub group also determines its behavior. Drawing a parallel with the platform strategy defined by Gawer and Cusumano (2002) and Gawer and Cusumano (2008), the identification of sub groups and their characteristics is pivotal to maintain a stable ecosystem.

Mathematicians and social scientists alike produced mechanisms to detect and study sub groups or communities in networks to uncover 'sub structures' underneath the general network topology. Algorithms can roughly be divided in two distinct approaches, being 'bottom-up' and 'top-down' (Hanneman & Riddle, 2005). Bottom-up algorithms regard the general network topology as emerging out of interrelated sub groups. Examples include cliques, n-cliques and clans (Hanneman & Riddle, 2005), that all start sub group detection from an adjacency pair outwards. On the contrary, top-down approaches take the bigger network as a starting point to decompose it into sub groups, in order to identify sparseness and weak spots in the ecosystem (Hanneman & Riddle, 2005; Scott, 2000). In subsequent analysis, only top-down approaches are applied as this research takes the total platform ecosystem as a starting point. Also, bottom-up approaches may be disturbed by the hub-and-spoke network topology of a platform ecosystem in which every new entrant has a relationship with the platform owner. In selecting suitable network metrics, optimization for larger networks is used as the main criterion.

One of the algorithms to divide a network into sub groups is modularity (Blondel, Guillaume, Lambiotte, & Lefebvre, 2008). Modularity is a network level metric, it results in high values when sub groups are densely connected while the relationships between sub groups are scarce. The algorithm searches for sparseness within a graph and detects communities (i.e. sub groups) accordingly by assigning each node to a modularity class. The algorithm is optimized for large networks, and is preferred because of its fast computation time (Blondel et al., 2008). Formalized, modularity is defined as follows:

Modularity: The degree to which a network can be divided into sub groups that are densely connected to each other, while sparsely connected to actors belonging to other sub groups.

A related measure that can also be operationalized on the node level is the clustering coefficient (Watts & Strogatz, 1998). The clustering coefficient of an actor is determined by the extent to which the connections of an actor are also connected with each other. The clustering coefficient of Google as a platform owner, for example, increases as the network density in the overall ecosystem rises. The clustering coefficient for the network as a whole, is the average of the clustering coefficient of all individual actors, and is an indicator for the cohesion in the ecosystem.

Clustering coefficient (CC_i): The ratio of the number of adjacencies of an actor that is also connected to one another to the total number of adjacencies an actor has.

To formalize the definition of local clustering coefficient (i.e. for a single actor), let K_i denote the number of adjacencies of node V_i. As the graphs are undirected, these adjacencies can initiate at most $K_i(K_i\text{-}1)/2$ interfirm relationships. If e_i is the observed number of interfirm relationships present between K_i neighbors, then the fraction with e_i in the numerator and enumerator of $K_i(K_i\text{-}1)/2$ results in the local clustering coefficient (Watts & Strogatz, 1998). The equation is shown in Equation 4.8. CC_i equals 0 if an actor has less than two adjacencies. The computation of the average clustering coefficient for the entire network is formalized in Equation 4.9

$$CC_i = \frac{2e_i}{K_i(K_i - 1)} \tag{4.8}$$

$$CC = \frac{1}{n}\sum_{i=1}^{n} CC_i \tag{4.9}$$

The application of measures to identify sub groups in ecosystem or business network literature are scarce. An exception to this is the work on proprietary ecosystems by Iyer et al. (2006). The authors use clustering coefficient to measure connectivity among partners in the ecosystems of large software companies. Among others, the authors argue that the clustering coefficient in the software industry is relatively high due to the intensive exchange between software companies, opposed to agriculture in which firms predominantly operate by themselves. Furthermore, the authors argue that the clustering coefficient around domain-specific platforms is higher, due to increased specialization, which in turn increases the need for collaboration.

4.4 Ecosystem Comparison

The vast majority of (social) network research are case studies of a single group, community or industry. Nevertheless, network comparison studies are a principal part of research on network topologies, as they provide means to compare networks across time and space. Most commonly applied are comparisons of two networks over the same set of actors. One such scenario is the comparison of two or more different relationships over the same set of actors (i.e. multiplex networks), as illustrated by Hanneman and Riddle (2005). Pattison and Wasserman (1999) propose a method for multivariate statistical comparison of 'multigraphs' akin to OLS regression.

Scarcer, yet more applicable to the comparison of platform ecosystems, is the application of replication as a comparison strategy. Replication arises when the same or a roughly comparable relation is measured over multiple sets of actors (Faust & Skvoretz, 2002). Examples of applied replication on alliance networks include the study of proprietary ecosystems by Iyer et al. (2006) and the comparison of industry alliance networks by Rosenkopf and Schilling (2007). Both studies contrast alliance networks based on their network metrics.

More systematic or statistical approaches towards replication are all based on p* models, also known as exponential random graph models (ERGMs) (Robins, Pattison, Kalish, & Lusher, 2007). An ERGM can be used to simulate networks that exhibit similar structural properties from an observed network. The approach is based on estimation and probability distributions that determine the extent to which a vector of network metrics, ties, triads or node characteristics in a graph influences the overall network topology. ERGMs allow statistical inference about networks (Robins et al., 2007) and can be used to contrast multiple networks, either by direct comparison of the ERGMs or by means of further estimation.

Anderson, Wasserman, and Crouch (1999) propose an extension to p* models to compare multiple networks and illustrate this method by contrasting the networks of friendship ties among children in different primary school classrooms. Since the authors argue computing p* models for each of the classrooms leaves little room for constraints, matrix aggregation is proposed as a way to compute a single p* model for multiple networks. Prerequisite herein is that an actor can only be present in one network and relationships are not recorded between actors in different networks. Consequently, the method proposed by Anderson et al. (1999) cannot be applied to compare platform ecosystems due to the prevalence of 'multi-homing' (Burkard et al., 2011) across software ecosystems. The same actors can thus be found in multiple networks, violating one of the prerequisites for network comparison in the defined method. In addition, matrix aggregation and p* computation of matrices that consist of hundreds of rows each becomes a tedious task.

Faust and Skvoretz (2002) propose a method inspired by p* models to compare networks that differ in size, type of relationships and species. The authors use probability distributions based on p* models to predict the network structure of other networks, followed by a correspondence analysis. The method is applied to the comparison of 42 networks that consist of between seven and more than 100 nodes. However, to the knowledge of the author the method has yet to be replicated by others, and its applicability to large networks so remains unclear.

In his work, J. Martin (1999) proposed a 'Data Analysis for Multiple Networks' framework that automated the comparison of multiple networks. While interesting, the framework never saw further development and thus cannot be applied to contrast platform ecosystems. Recent advances in the same line of research have resulted in the creation of the Statnet[1] package for the open source statistical language R. Its application to date is limited to simulation of networks based on ERGMs, and thus provides limited means to contrast multiple networks. Accordingly, subsequent contrasting of platform ecosystems will be performed through the comparison of network visualizations and their respective metrics, due to the sheer availability of viable and readily implementable alternatives.

[1] http://statnet.csde.washington.edu

Chapter 5

Case Study Results

Whereas the preceding chapters provided the theoretical fundament of this thesis, this chapter describes the results of four case studies on platform ecosystems. Google and Microsoft are selected as case companies because of the large overlap in their platform portfolio. For Google, Google Apps and Google Chrome are chosen as subject of study, and for Microsoft Office365 and Internet Explorer were selected. The selection process proceeded in accordance with the selection criteria as defined in Chapter 2. Table 5.1 provides an overview of start dates for the data collection for each case study. The app store data extraction for the web browsers started later because the web crawler for Google Apps and Office365 was developed first.

Table 5.1: Start dates for the data collection process per platform

Platform	**Start date**
Google Apps	12-02-2013
Office365	12-02-2013
Google Chrome	27-05-2013
Internet Explorer	27-05-2013

The remainder of this chapter elaborates upon the case study results per platform. Case study results are described per case company, because the characteristics of this company may be used in interpreting case study findings. Cross-case comparison will only be presented in the next chapter, to maintain strict separation of case result description and analysis. In addition, it allows the interpretation of a staggering amount of case study data and focus on within-case patterns, rather than more generic patterns that can be identified by means of cross-case analysis (Eisenhardt, 1989). The Google platforms are elaborated upon first, succeeded by a description of the Microsoft platform ecosystems.

5.1 Google

Google is a multinational for-profit corporation that was founded in 1998. Google specializes in internet-related products and services. Its product portfolio among others consists of a mobile operating system, search engines, advertisement services and an array of cloud applications. Google is perceived as a relatively 'open' company, due to its transparency, intensive use of open source software and the limited entry barriers imposed for potential developers of complements. Since recently, Google operates its Enterprise Partner[1] partnership model, targeted at participation of any enterprise that actively affiliates with Google. Furthermore, Google employs certification programs for resellers and service providers for specific Google platforms.

5.1.1 Google Apps

Google Apps is a cloud based office suite platform, intended for use by enterprises and governmental or educational institutions. The platform consists of customizable versions of Google products, that include Gmail, Google Drive, Google Sites and Google Calendar. Google Apps depends on complements to extend or diversify the existing platform functionality. Examples of complements are data migration tools, integrations with other platforms and applications that make Google Apps suitable to for instance perform customer relationship management. Google offers transparently documented APIs and SDKs for prospective developers, and provides dedicated support sources on its Google Developer portal. The complements – either developed by Google or third-parties – can be found in the Google Apps Marketplace[2]. Listing applications in the marketplace can be done free of charge, however, all applications are subjected to a validation process in before formal approval is granted.

On *12-02-2013*, the Google Apps Marketplace contained a total of *1354* applications listed under the category *'Products'*, developed by *993* different vendors or individuals. Out of the 1354 applications, *thirteen* are developed by Google, using the aliases *Google Inc.* and *Google Labs*. The *992* remaining complementors develop *1341* applications, with contributions varying between *one and fifteen* applications. On average, each actor develops *1.36* complements with a standard deviation of *0.61*, showing that the vast majority (83%) of the ecosystem consists of complementors that develop one application. A complete overview of vendor distribution and the number of complements they develop is shown in Table 5.2.

[1] http://www.google.com/enterprise/gep
[2] http://www.google.com/enterprise/marketplace

Table 5.2: Distribution of Google Apps complementors based on the number of applications developed

# of applications	# of complementors
15	1
14	2
11	1
10	1
9	2
8	2
7	4
6	5
5	6
4	10
3	26
2	104
1	826

Google Apps complementors limitedly participate in the partnership model and certification programs employed by Google. To date, *71 (7.16%)* complementors obtained certification for reselling services or additional service provision, while *fourteen (1.41%)* vendors are currently listed Google Enterprise Partner. Participation in both programs accounts for a mere *7.36%* of the total ecosystem, as *73* vendors participate in one or both programs.

According to the data collected from company websites and CrunchBase, the ecosystem is connected by *1248* visual relationships. Actors are connected to one another with an average of *1.26* relationships per actor, which results in a sparsely connected hub-and-spoke network. This hub-and-spoke network is shown in Figure 5.1. Node sizes in the figure are representative for the number of applications developed. The network level descriptives of this figure are summarized in Table 5.3.

Table 5.3: Network metrics for the Google Apps ecosystem

Metric	Value
Size	993
Network density	0.00253
Centralization	0.9995
Modularity	0.278
Clustering coefficient	0.734

As shown in Figure 5.1 there is a limited number of lateral connections among complementors in the Google Apps ecosystem, the majority of which is concentrated in the bottom right of the figure. Following a centralization score of *99.94%*, Google is regarded as the central player responsible for employing the initiatives in the ecosystem. With a network density that is lower than *1%*, however, the Google Apps ecosystem has a sparse

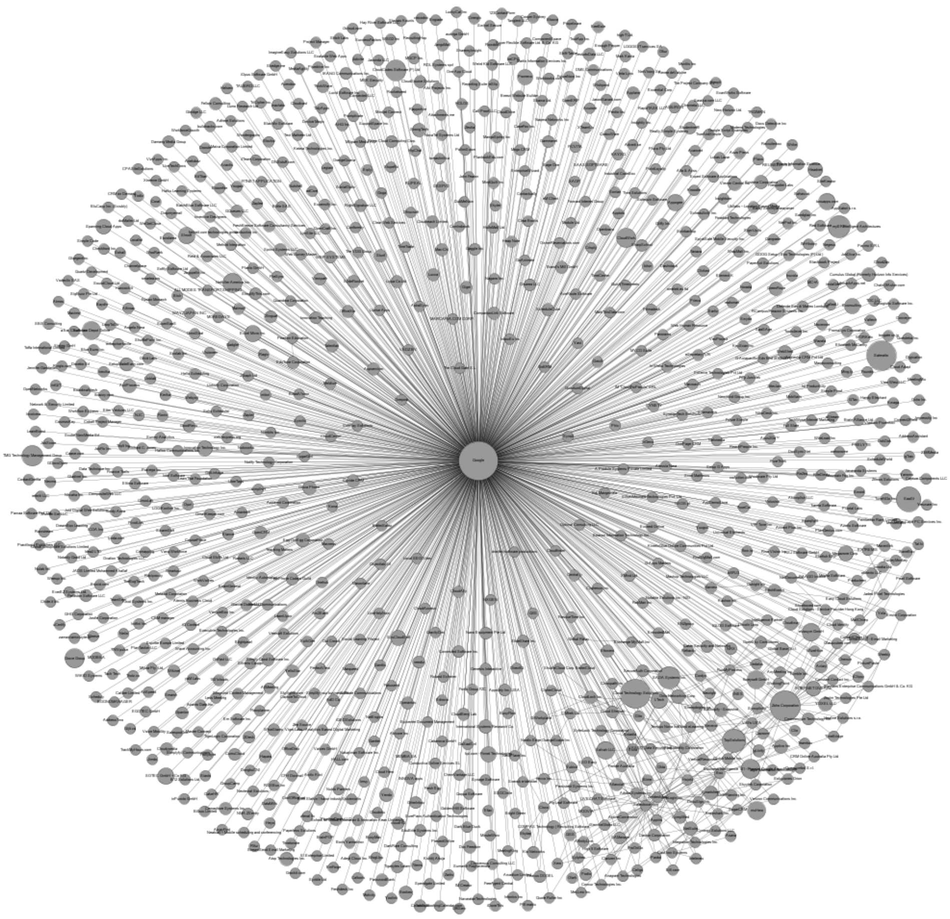

FIGURE 5.1: Visualization of the Google Apps ecosystem

network structure. The majority of complementors are thus new entrants, individuals rather than enterprises, limit themselves to the development of their own applications, or maintain a selective set of key relationships rather than engaging in multiple interfirm relationships.

The node level metrics for the Google Apps ecosystem are summarized in Table 5.4. The high standard deviation for degree centrality on the node level reveals the presence of a small group of influential actors in the ecosystem, *73%* of actors are solely connected to Google. Closer inspection of the dataset confirms this finding, identifying *Salesforce.com* as the most embedded actor with *35* relationships, followed by *Zoho Corporation* with a relationship count of *20*, and *Box* that has *20* relationships after which comes *Ping Identity Corporation* with *16* relationships. The Bonacich power index returns a differing sequence of complementors. While *Zoho Corporation* with a power index of *2.03* still is among the powerful actors in the ecosystem, *Salesforce.com* is not found in this list. Subsequently, partners of *Salesforce* are relatively well embedded in the ecosystem, compared to the partners of *Zoho Corporation*. Accordingly, Zoho Corporation could be considered a pivotal connector in the ecosystem.

Table 5.4: Network metrics for the Google Apps ecosystem

Metric	Min.	Max.	Avg.	Std. dev.
Degree centrality	0.00101	1	0.00253	0.0317
Eigenvector centrality	0.0320	0.0670	0.0320	0.00100
Bonacich power (β = -0.5)	-2.10	2.05	2.00	0.00100
Clustering coefficient	0.000491	1	0.734	0.287

The relatively high values for modularity and clustering coefficient reveal the presence of clusters in the Google Apps ecosystem. Accordingly, the dataset was 'cleansed' to provide additional insights. First, to uncover the topology beneath the hub-and-spoke network, Google has been removed from the visualization. Second, complementors solely connected to Google have not been included in the graph. The resulting network visualization of the Google Apps ecosystem is shown in Figure 5.1.

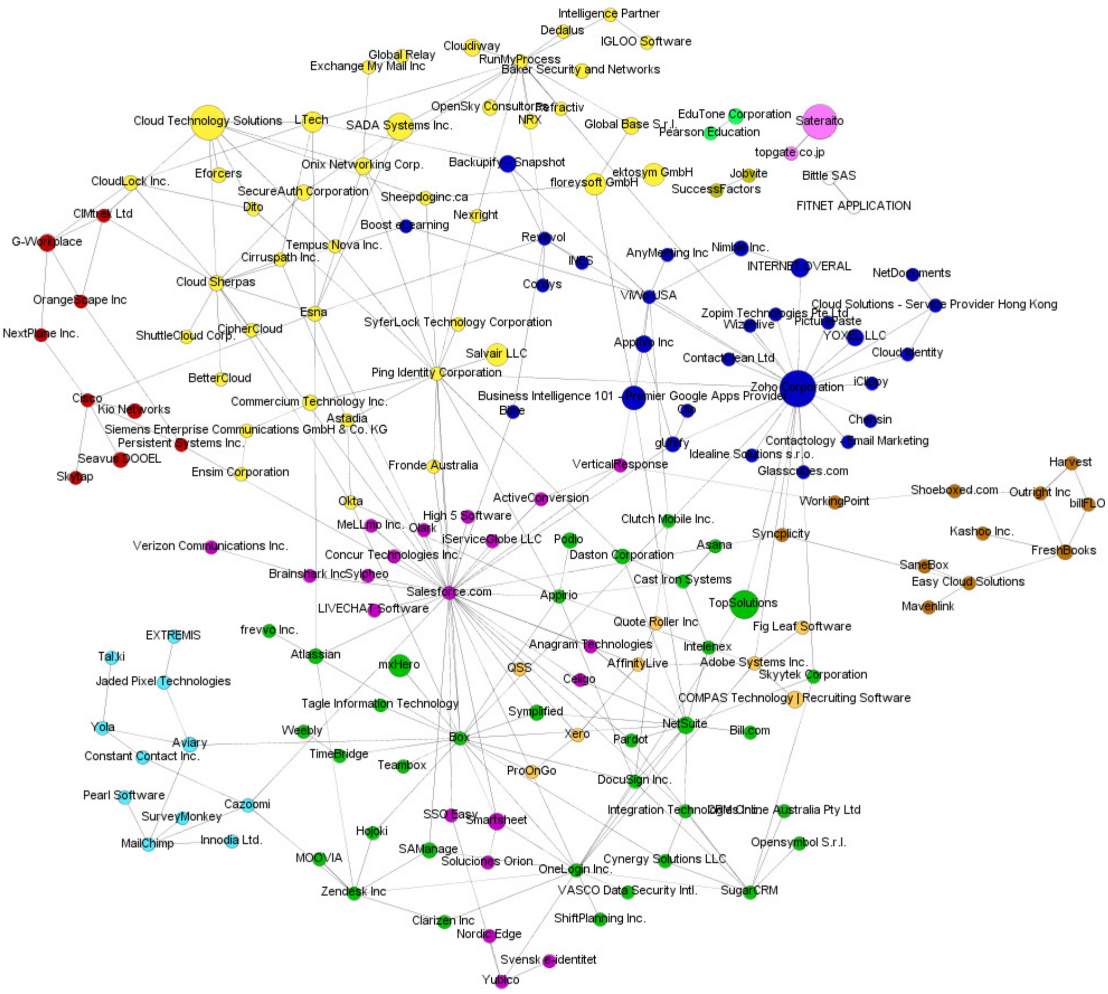

Figure 5.2: Cluster visualization of the Google Apps ecosystem

In total, *twelve* clusters were found of which the members are grouped accordingly in Figure 5.2. Observable is the difference in cluster sizes. There are four clusters that consist of two complementors that interact with each other, but not with the other actors in the ecosystem. The small clusters consist of companies that are headquartered at the same geographic location. *Sateraito* and *topgate.co.jp*, for example, are both Japanese whereas *Bittle SAS* and *FITNET APPLICATION* are French. The larger clusters reflect more differences. The *Salesforce.com cluster* for instance has a hub-and-spoke network topology, due to the platform efforts of *Salesforce* itself. Similarly, companies from different geographic locations are grouped around *Zoho Corporation*, making the clusters of *Salesforce.com* and *Zoho Corporation* examples of technology clusters. Other clusters, such as the one in the upper left corner with *OrangeScape* in the middle, lack one central actor and seem to be merely groups of actors connected to one another through direct and indirect relationships. Geographic location herein does not seem to be an influential factor, as the *OrangeScape cluster* includes multinationals and companies headquartered in the United States, Canada, Mexico, Sweden and the Netherlands.

Noteworthy is the absence of a couple of large actors within the ecosystem. *SaaSt* for instance at present offers eleven applications for Google Apps, yet they are not connected to any other complementor in the ecosystem. Similarly, *myERP* is absent while it currently lists nine applications in the Google Apps Marketplace. Other large complementors such as *Zoho Corporation* (15 applications), *Cloud Technology Solutions* (14 applications) and *TopSolutuions* (10 applications) are among the most central actors in the ecosystem.

5.1.2 Google Chrome

Google Chrome is a web browser that saw its first launch by Google in 2008. Extensions for Google Chrome include web versions of commonly used applications, games, toolbars, RSS feeds, developer tools and utilities. Extensions can be found in the Google Chrome Web Store[3]. Of all available complements, this research only considers the ones listed as *'extensions'*, because other categories predominantly include links to websites or web versions of applications. The extension architecture for Google Chrome was introduced in 2009 and is comprised of APIs for specific 'parts' of Chrome, each of which is documented on the Chrome Developer Portal. In addition, Google freed part of the source code of Chrome, which is also used in the ongoing Chromium project. Apart from a minor first-time publisher fee($10), Google does not impose any (quality) constraint to publishing complements in the Google Chrome Web Store.

The Google Chrome Web Store contained a total of *2057* extensions on *27-05-2013*, which are developed by *1540* complementors. Complementors on average develop *1.34*

[3]http://chrome.google.com/webstore

extensions with a standard deviation of *0.59*, an indication of the presence of a large number of small developers in the ecosystem. This finding is supported by the vendor distribution shown in Table 5.5. Google is active in entering complementary markets and at present lists *52* Google Chrome extensions.

TABLE 5.5: Distribution of Google Chrome complementors based on the number of applications developed

# of applications	# of complementors
52	1
39	1
33	1
27	1
24	2
10	2
8	2
7	2
6	3
5	8
4	15
3	35
2	118
1	1349

Complementors in the Google Chrome ecosystem are scarcely connected. The members of the ecosystem are connected by *1586* visible interfirm relationships. On average, each complementor engages in *1.03* relationships with a standard deviation equal to *19.59*. Of all complementors, only *3.18%* collaborate or integrate with another actor in the ecosystem, as 49 companies make mentions of partnerships on their company website. Consequently, the rounded centralization of the Google Chrome ecosystem equals *100%*, implying that Google is the sole responsible actor for initiatives and diffusion thereof in the ecosystem. The Google Chrome ecosystem is visualized in Figure 5.3, its network level metrics are summarized in Table 5.6, node level metrics are shown in Table 5.7.

TABLE 5.6: Network metrics for the Google Chrome ecosystem

Metric	Value
Size	1540
Network density	0.00100
Centralization	1.00
Modularity	0.056
Clustering coefficient	0.849

The limited interconnectivity in the ecosystem combined with the large number of small complementors support the hypothesis that the ecosystem is populated by actors such as individuals and small enterprises or that there is only limited commitment to the platform by third-parties. The second hypothesis is supported by the activity of Google in entering complementary markets, whereas additional evidence is needed for the first proposition. Therefore, all complementors were categorized based on their organizational form. The categorization is based on information found in the vendor description in the Google Chrome Web Store and information retrieved from web pages. A distinction is made between the following types of complementors:

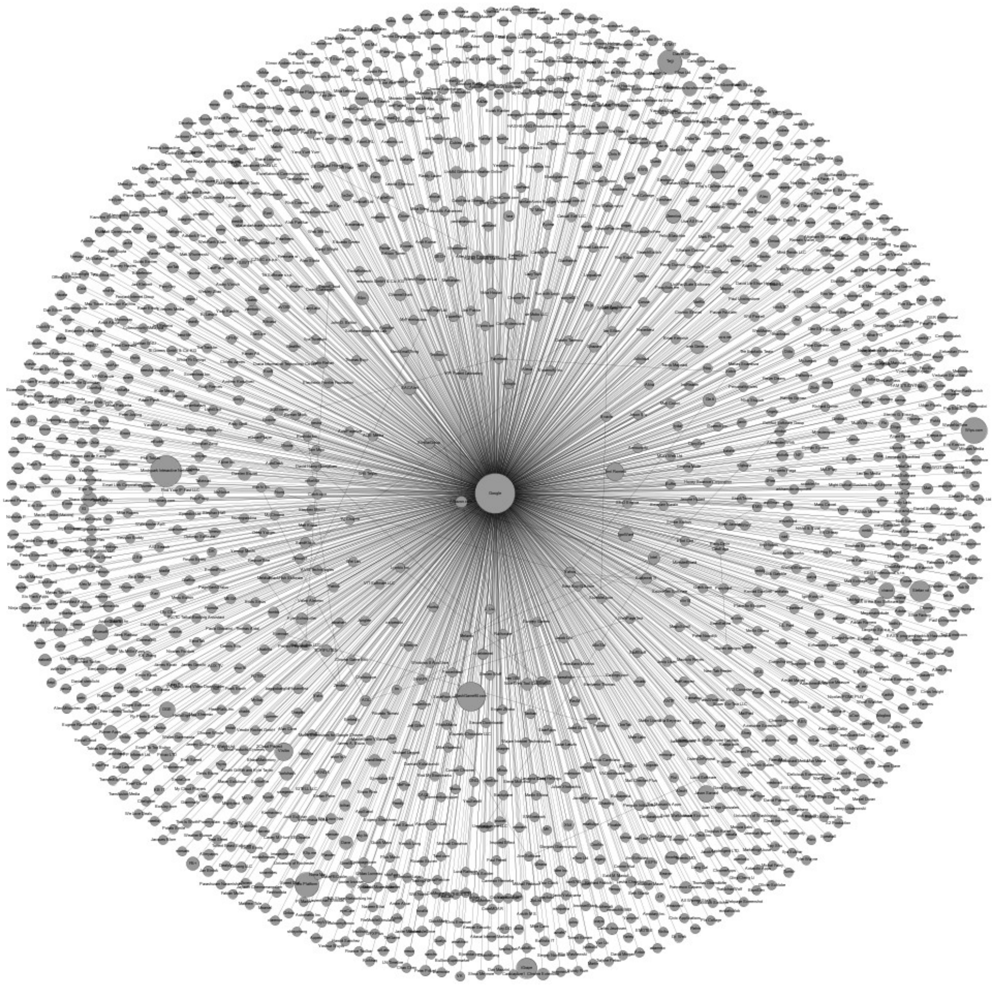

FIGURE 5.3: Visualization of the Google Chrome ecosystem

TABLE 5.7: Network metrics for the Google Chrome ecosystem

Metric	**Min.**	**Max.**	**Avg.**	**Std. dev.**
Degree centrality	0.00065	1	0.00134	0.0255
Eigenvector centrality	0.0180	0.0707	0.0180	0.00100
Bonacich power (β = -0.5)	-6.00	-2.00	-2.00	0.00100
Clustering coefficient	0.000118	1	0.849	0.205

- **Community/Non-Profit:** A non-profit organization or a group of individuals that collectively develop complements, recognizable by links to community forums, open source projects or source code bases such as Google Code[4], SourceForge[5] and GitHub[6].

- **Company:** For profit organization identified by means of company websites or company suffixes such as Inc., LLC, B.V. and GmbH.
- **Educational Institution:** University.
- **Individual:** A single developer that uses a first or last name in its complementor alias or links to its personal website or portfolio.
- **N/A:** Complementor for which it is impossible to determine to what category they belong.

The ecosystem was found to be populated by *135 (8.77%* non-profit organizations, *490(31.84%)* companies, *five(0.32%)* educational institutions, *637(41.39%)* individuals and *272(17.67%)* actors for which it was not possible to determine their type. Noteworthy is the large number of individuals that contribute to the development of extensions for Google Chrome. As these complementors typically develop opportunistically (Hyrynsalmi, Suominen, Makila, Jarvi, & Knuutila, 2012) and do not establish relationships among each other – apart from participation in open source communities which is outside the scope of this research – the number of actors that potentially exchange among each other dramatically decreases. From the companies, *Amazon.com* is most well embedded with *thirteen* interfirm relationships, followed by *Yahoo* that has *six* connections and *Intel* and *Raven Internet Marketing Tools* that both engage in *five* partnerships. The cluster visualization of interacting businesses in the ecosystem is presented in Figure 5.4.

The interacting complementors in the Google Chrome ecosystem can be divided in twelve clusters, six of which are constituted of two complementors. Meanwhile, only the *Meldium*, *Amazon.com* and *Raven Internet Marketing Tools* clusters are mutually connected through lateral connections. This overall lack of connectivity indicates the presence of functional clusters in the ecosystem. Illustrative is the left side of the *Raven Internet Marketing Tools* sub group, it consists of internet companies that concentrate on social media, search engine optimization and marketing. In addition, it is an indicator that the identified interfirm relationships cannot be solely attributed to exchanges directly related to the Google Chrome ecosystem. This finding is supported by the lack of interfirm relationships between the largest vendors in the ecosystem. Apart

[4]http://code.google.com
[5]http://www.sourceforge.net
[6]http://www.github.com

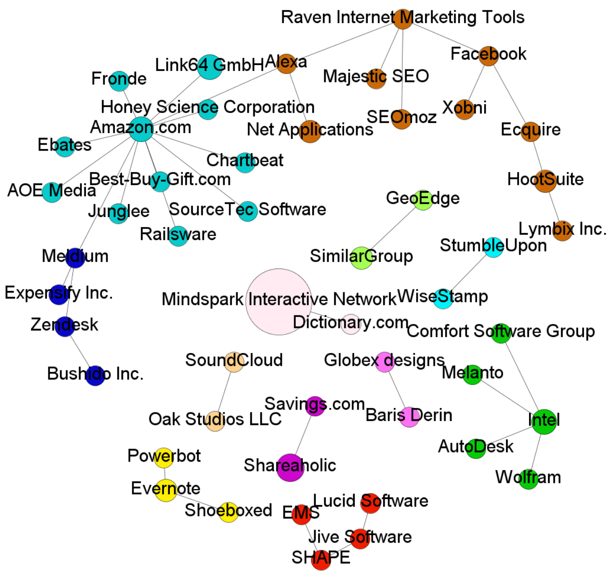

FIGURE 5.4: Cluster visualization of the Google Chrome ecosystem

from *Mindspark Interactive Network*, none of the thirty-five largest actors mention any interfirm relationship with other Google Chrome complementors.

5.2 Microsoft

Founded in 1975, Microsoft is a multinational for-profit corporation that operates in the wider computing industry. The Microsoft product portfolio consists of both on-premises and cloud based products, it includes PC and mobile operating systems, an office suite, a search engine and a video gaming console. Microsoft employs numerous partnership models and certification programs that are at the core of their business. The programs are organized under the 'Microsoft Partner Network'[7] umbrella. Microsoft is perceived to be relatively closed, due to its numerous proprietary standards, contained governance strategy and proprietary licensing agreements.

5.2.1 Microsoft Office365

Microsoft Office365 is a cloud based office suite platform. For consumers, it merely provides access to cloud versions of Microsoft Office products, whereas the business versions of Office365 offer additional services for small businesses, enterprises, and governmental or educational institutions. For professional use, Office365 bundles scalable versions of Microsoft Office, Microsoft Lync, Microsoft Exchange and SharePoint. Complements for

[7]http://www.mspartner.microsoft.com

Microsoft Office365 can be found on a dedicated part of the bigger Microsoft Pinpoint marketplace[8]. Examples of complements range from integrations and add-ons to business templates for Microsoft Office products. Complements can be either listed globally *(45.5% of the applications)*, or can be listed in one of the fifty-nine regional marketplaces.

In order to be listed on the Office365 Marketplace, applications have to be subjected to compatibility, certification and complementary value requirements defined by Microsoft. At present, Microsoft does not provide an API for Office365. Instead, all components of Office365 offer their own extension mechanism. Microsoft Exchange and Lync have ready-to-use APIs, while Microsoft offers SDKs for complementors that focus on extensions for Sharepoint or Office.

TABLE 5.8: Distribution of Office365 complementors based on the number of applications developed

# of applications	# of complementors
39	1
32	2
23	1
22	1
16	1
14	1
13	1
11	3
10	3
9	5
8	4
7	7
6	13
5	10
4	23
3	24
2	82
1	368

At *12-02-2013*, *550* complementors developed one or more applications for Microsoft Office365, listed in either the global or one of the regional marketplaces. Of the 550 complementors *278 (50,50%)* are Microsoft partners. Noteworthy is that Microsoft itself does not actively enter complementary markets for Office365. Accordingly, all *1204* applications are developed by third-parties. On average, each complementor develops *2.18* applications with a standard deviation of *1.65*. The largest vendor in the ecosystem lists *39* applications, while *67%* of complementors only develop one application. A complete distribution of these figures is included in Table 5.8. Important to note is that the distribution of complementors may be influenced by Microsoft listing each extension

[8]http://www.office365.pinpoint.microsoft.com

for a Microsoft Office component as a separate complement. When a complementor, for example, develops an application that can work with Microsoft Word, Excel and Powerpoint, it can be found three times in the Office365 Marketplace. Subsequently, the actual number of unique complements for Office365 may be lower than the number found in this research.

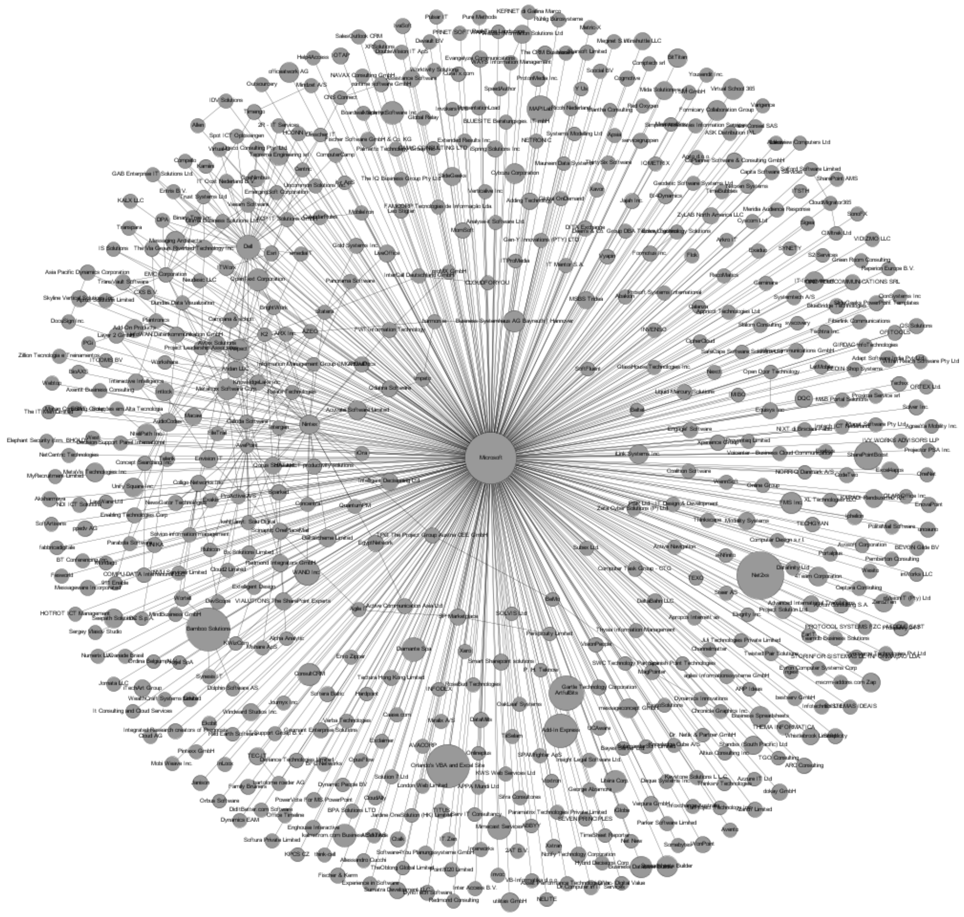

FIGURE 5.5: Visualization of the Office365 ecosystem

TABLE 5.9: Network metrics for the Office365 ecosystem

Metric	Value
Size	551
Network density	0.00500
Centralization	0.9984
Modularity	0.336
Clustering coefficient	0.773

The Office365 complementors are connected by *787* relationships, accounting for an average of *1.43* relationships per complementor with a standard deviation of *11.74* relationships. The most well connected actors are *Dell* with *37* relationships and *Nintex* with *32* relationships that are among the largest vendors in the ecosystem, followed by the small vendors *AvePoint* and *K2* with *23* and *16* relationships. The network density equals a value of *0.5%*. Consequently, the centralization score for the Office365 ecosystem is *99.84%*, lower than the previously discussed ecosystems. The network graph of the entire network is shown in Figure 5.5, the descriptives of which are included in Table 5.9 and Table 5.10.

TABLE 5.10: Network metrics for the Office365 ecosystem

Metric	Min.	Max.	Avg.	Std. dev.
Degree centrality	0.00183	1	0.00519	0.0427
Eigenvector centrality	-0.0320	0.0670	0.00200	0.0430
Bonacich power (β= -0.5)	-2.02	2.06	2.04	0.00100
Clustering coefficient	0.00215	1	0.773	0.228

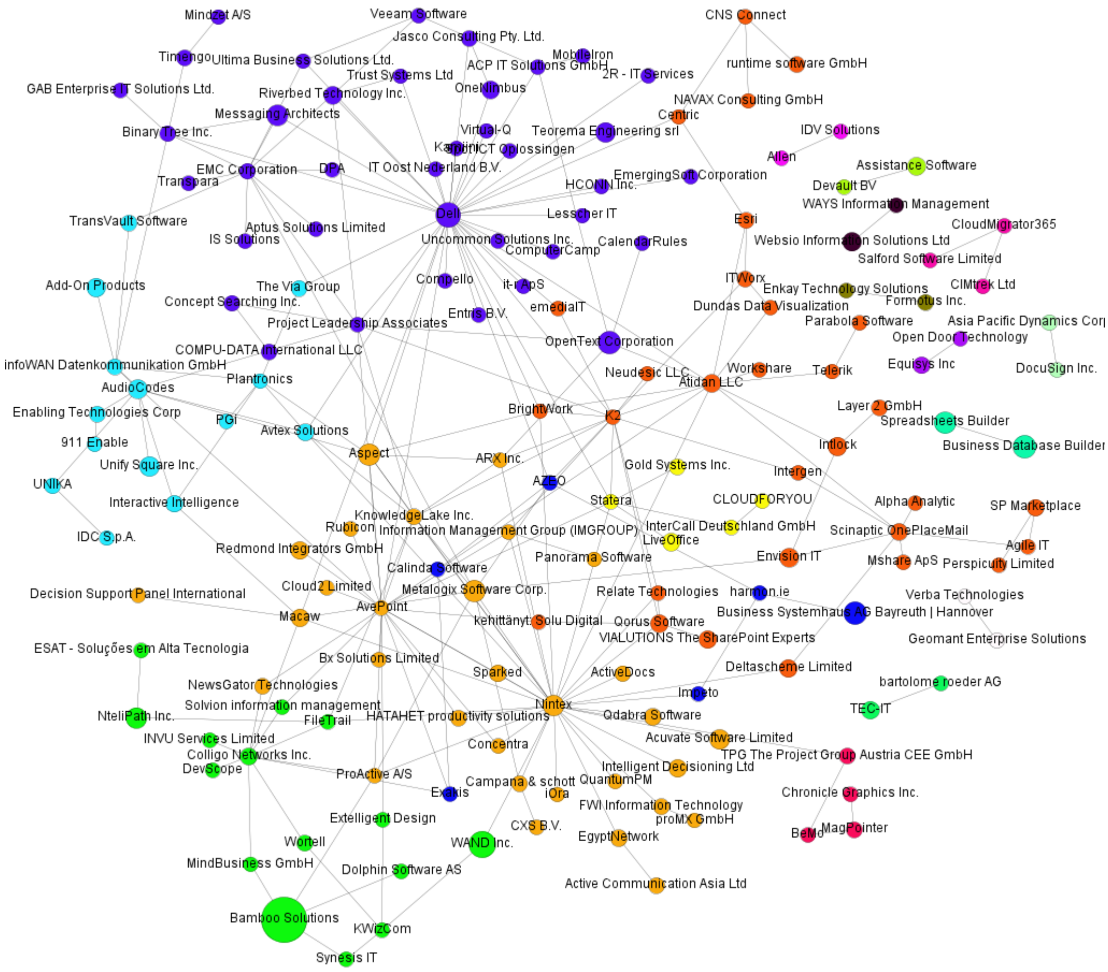

FIGURE 5.6: Cluster visualization of the Office365 ecosystem

Noteworthy is that some of the largest vendors in the Office365 Marketplace seem to be individual developers or one man businesses. The second largest vendor in the ecosystem for instance is called *Orlando's VBA and Excel Site* and at present lists *32* applications. Inspection of the web page associated with this vendor reveals that the developer is an individual, dedicated to Microsoft Office development and support. Despite the individual developers in the ecosystem, the modularity score and clustering coefficient are relatively high, indicating interconnectedness among complementors. This interconnectedness, may be partially explained by the presence of established Sharepoint developers in the ecosystem, as the on premises version of the product already came with an extension architecture. In total, 164 out of 550 complementors *(29,82%)* engage in partnerships with other Office365 complementors. These 164 complementors are distributed over *eighteen* clusters, shown in Figure 5.6.

The network topology in Figure 5.6 is constituted of either large or very small clusters. The largest cluster in the ecosystem is visualized in the upper left corner of the graph, showing the complementors that have intensive technological ties with *Dell*. These complementors, however, do not seem to intensively interact with each other, opposed to the actors that surround *Nintex* and *AvePoint*. The *Nintex* cluster seems to embody the most interconnected part of the ecosystem, populated by relatively large vendors.

5.2.2 Internet Explorer

Internet Explorer is a web browser that was first launched by Microsoft in 1995. The browser has a modular architecture that is extensively documented on a dedicated website. The component-based architecture enables third-parties to extend part of the web browser by means of Component Object Model interfaces. These interfaces allow third-party extensions to access part of the functionality of the platform. Add-ons can be found in the Internet Explorer Gallery[9] and include toolbars, accelerators, parental control add-ons and news or content feeds. In before inclusion in the Internet Explorer Gallery, add-ons are subjected to a review process by Internet Explorer developers. Apart from add-ons, Internet Explorer Gallery also include 'pinned sites' – that brand the user interface of Internet Explorer in styles of popular web pages such as Facebook, last.fm or The New York Times – which were excluded from consideration in the data collection process.

On *27-05-2013* the Internet Explorer Gallery contained *853* add-ons developed by *518* complementors. Most prominent complementor is *Brand Thunder LLC* that developed *80* add-ons for third-parties such as professional sports clubs and popular brands. Contrary to the Office365 ecosystem, Microsoft actively penetrates complementary markets in the Internet Explorer ecosystem. It at present develops *22* add-ons. On average, each complementor develops *1.64* add-ons with a standard deviation of *1.01*, *78.57%* develop one complement. A complete distribution of complementors is shown in Table 5.11.

As shown in Table 5.11 only fifteen complementors develop more than five add-ons for Internet Explorer, reflecting sheer commitment to the platform. To verify if this

[9] http://www.iegallery.com

Table 5.11: Distribution of Internet Explorer complementors based on the number of applications developed

# of applications	# of complementors
80	1
22	1
11	1
10	1
9	2
8	3
7	2
6	4
5	3
4	9
3	20
2	64
1	407

implies the presence of a large number of individual developers in the ecosystem, a complementor type categorization was created. The categorization encompasses the same complementor types as those described in Section 5.1.2, with the addition of local governments. Multiple states (*eight (1.55%)*) in the USA developed their own web slices. Apart from governmental institutions, the ecosystem is populated by *50 (9.67%)* non-profit organizations, *323 (62.48%)* companies, *five (0.97%)* educational institutions, *58 (11.12%)* individuals and *73 (14.12%)* complementors of which their type remained unknown. Noteworthy is that contrary to the classification described for Google Chrome, the majority of complementors in the Internet Explorer ecosystem are companies rather than individuals.

The Internet Explorer ecosystem is connected by a total of *554* relationships. Each complementor on average initiated *1.07* interfirm connections with a standard deviation of *11.33* relationships. The most well embedded complementors are *CareerBuilder* with seven interfirm relationships, followed by *blinkx*, *Google* and *Amazon* all having six connection in the ecosystem. Figure 5.7 shows the network graph that represents the complete Internet Explorer ecosystem. The most densely connected parts of the ecosystem can be found just above and below Microsoft.

Table 5.12: Network metrics for the Internet Explorer ecosystem

Metric	Value
Size	518
Network density	0.00400
Centralization	0.9997
Modularity	0.117
Clustering coefficient	0.771

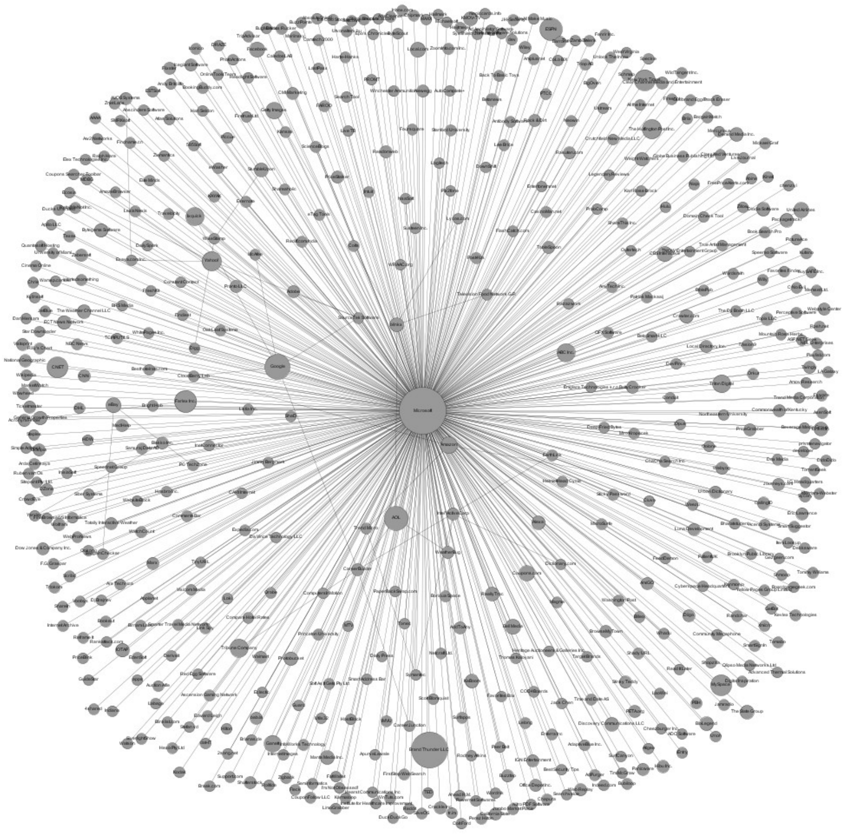

FIGURE 5.7: Visualization of the Internet Explorer ecosystem

Similar to the previously discussed ecosystems, the Internet Explorer ecosystem is highly centralized *(99.7%)* due to a network density that is equal to *0.4%*. This overall scarcity causes modularity to be low, hence there is not enough differentiation between densely and scarcely connected parts of the ecosystem. Consequently, Bonacich power indices are predominantly negative due to a lack of actors that bridge gaps too less densely connected parts of the ecosystem. A complete overview of network metrics for the Internet Explorer ecosystem is summarized in Table 5.12 and Table 5.13.

TABLE 5.13: Network metrics for the Internet Explorer ecosystem

Metric	**Min.**	**Max.**	**Avg.**	**Std. dev.**
Degree centrality	0.00193	1	0.00414	0.0438
Eigenvector centrality	0.0310	0.706	0.0320	0.00300
Bonacich power (β = -0.5)	-6.05	-1.93	-2.26	0.00500
Clustering coefficient	0.000133	1	0.771	0.246

At present, *38* complementors have established interfirm relationships among each other. Complementors can be divided in eight sub groups, two of which are populated by two complementors. As shown in the cluster visualization for the Internet Explorer ecosystem in Figure 5.8, the remaining clusters are generally interconnected by one or more intermediaries, referred to as 'bridges'. Interfirm relationships predominantly appear to exist between complementors with a similar specialization. Illustrative examples are the connections between anti-virus software developers *Symantec* and *Trend Micro*, media company *Television Food Network G.P.* and video sharing and search platform *blinkx*, recruiting platforms *Career Junction* and *CareerBuilder* or the interfirm relationship between *Google* and niche search engine of *lxquick*. Noteworthy is that nine out of twenty of the most active complementors in the Internet Explorer ecosystem are present in the cluster visualization. However, none of these nine actors are directly connected through an interfirm relationship.

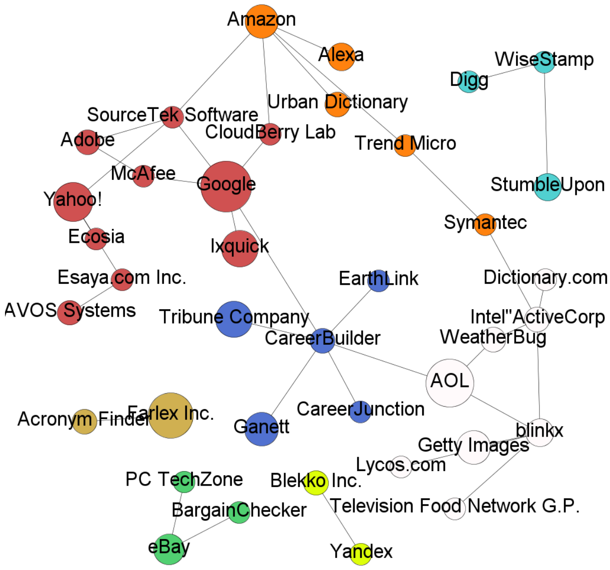

FIGURE 5.8: Cluster visualization of the Internet Explorer ecosystem

Chapter 6

Comparing Platform Ecosystems

This chapter presents the results a cross-case analysis of the four case studies that were described in the previous chapter. While the previous chapter elaborated upon each case in isolation, commonalities stand out. First of all, all platform ecosystems exhibit the same structural properties in their global network topology. The platform ecosystems are sparsely connected and highly centralized hub-and-spoke networks. Interfirm relationships are limitedly initiated among actors in the ecosystem, indicating a focused relational strategy of firms under increasing specialization or a lack of interest in initiating collaboration. Second, the majority of complementors in all ecosystems *(between 66.85% and 87.58%)* develop one application. The large number of small complementors could be explained by the relative novelty of online marketplaces with complementarities or the ongoing search for viable business models to exploit the new app industry paradigm. These circumstances give rise to increased opportunism in the ecosystem, certainly as start-up investments are next to zero (Butler, 2011; Hyrynsalmi, Suominen, et al., 2012). Last, each platform seems to have imposed at least an extent of entry barriers to inclusion of new complementarities in the app store.

Despite the apparent similarities, the four platform ecosystems can also be contrasted to identify noteworthy differences. Table 6.1 summarizes the principal descriptives for each of the four platform ecosystems. Metrics such as global clustering coefficient have not been included in Table 6.1, because these are likely to be disturbed by the global hub-and-spoke network topology. As a consequence, the clustering coefficient will hardly differ across the four platform ecosystems. Instead, the clustering coefficient of the platform owner is included, because it is an indicator of interconnectivity among complementors that is not disturbed by the hub-and-spoke network topology (Iyer et al., 2006).

Table 6.1: Descriptives of the four platform ecosystems

Characteristic	**Google Apps**	**Google Chrome**	**Office365**	**Internet Explorer**
Owner	Google	Google	Microsoft	Microsoft
Platform type	Office suite	Web browser	Office suite	Web browser
Entry barriers	Technical validation	Minor first-time publisher fee	Compatibility and complementarity value validation	Technical validation
Ecosystem size	993	1540	551	518
Avg. applications per actor	1.36	1.34	2.18	1.64
Standard deviation for avg. applications	0.61	0.59	1.65	1.01
Actors with one application	83.18%	87.58%	66.85%	78.57%
Avg. relationships per actor	1.26	1.03	1.43	1.07
Standard deviation for avg. relationships	15.74	19.59	11.74	11.33
Avg. relationships per actor (excl. platform owner)	0.258	0.0305	0.431	0.0715
Standard deviation for avg. relationships (excl. platform owner)	1.01	0.225	1.37	0.317
Actors with relationships	17.14%	3.18%	29.82%	7.35%
Network density	0.25%	0.010%	0.50%	0.040%
Centralization	99.94%	100%	99.84%	99.97%
Clustering coefficient platform owner	0.000491	0.000118	0.00215	0.000133

The descriptives summarized in Table 6.1 will be used for the cross-case analysis presented in this chapter. Cross-case analysis aims for the synthesis of one or more theories in an iterative process (Eisenhardt, 1989). Contrary to the approach described by Eisenhardt (1989), propositions rather than theories are formulated in this chapter. The lack of means to perform statistical comparison of networks that differ in size, nature and species poses limitations to the reliability of a resulting theory. The presented propositions are interpreted as a formalization of observed effects, and as such need statistical interference in future work to enhance their reliability and generalizability. The analysis is formulated around the last two of four sub research questions that were defined as part of the introduction in Chapter 1. The sub research questions were formulated as follows:

- **RQ 3:** How do the characteristics of the firm ecosystem influence the network structure of its containing platform ecosystems?
- **RQ 4:** What is the influence of the type of platform on the network structure of the platform ecosystem that exists around it?

The remainder of this chapter contrasts the four platform ecosystems from different perspectives. It addresses the perspectives of entry barriers, likelihood of interfirm relationship initiation, partnership model participation, species populating the ecosystem and complementor distribution as sources for potential differences among platform ecosystems. The literature review presented in Chapter 3 brought these mechanisms forward to have a potential influence on the network structure of ecosystems.

6.1 Entry Barriers

Openness and entry barriers are perhaps one of the most recurrently addressed aspects of platform strategy. In accordance with the perspective on platform openness by Eisenmann et al. (2009), Google and Microsoft platforms can be contrasted based on their degree of architectural control, the extension architectures in place and entry barriers to entering complementary markets. With regard to architectural control, Google Apps, Office365 and Internet Explorer platforms are considered comparably 'closed'. Because Google grants access to part of the source code of Google Chrome, it is perceived more easily accessible. With regard to entry barriers to entering complementary markets, the barriers imposed by Google appear more liberal than those of Microsoft. Google does not impose any quality requirement for complements to be included in the Google Chrome Web Store, apart from the payment of a minor first-time publisher fee. Google Apps and Office365 are comparable in their architectural requirements, however, Microsoft imposes discriminatory complementary value requirements. With these requirements in hand, Microsoft reserves the right to exclude any application from the app store that does not provide direct added-value to the platform.

Apparent from the ecosystem descriptives presented in Table 6.1 is the higher number of complementors for both Google platforms opposed to their Microsoft equivalents. Both Google ecosystems are larger than their Microsoft counterparts, with a difference of *440* complementors for the cloud office suite platforms and *1022* complementors for the web browsers. This higher rate of adoption coincides with more liberal entry barriers. More concrete, Google sees two to three times more complementors for their platforms with more liberal governance. Accordingly, empirical evidence postulates that lowering entry barriers to a platform ecosystem will be positively related to the number of complementors for a platform.

Proposition 1: Lowering entry barriers to a platform ecosystem will be positively related to the number of complementors.

This proposition finds supportive evidence in existing scientific literature. West (2003), Jansen, Brinkkemper, and Finkelstein (2009) and Cusumano (2010b) draw parallels between increased openness and increasing rates of adoption or ecosystem growth. With their 'open enterprise model', Jansen et al. (2012) envisage a need for increased openness by providing software producing organizations with a model to determine how 'open' or 'closed' organization is and guidelines to increase openness. The authors argue that product or platform selection are based on prejudice with regard to openness and transparency. As Jansen et al. (2012) reason, a lack of transparency and openness may harm the reputation of a platform owner and its ecosystem, for they are deemed to foster lock-ins and abuse of intellectual property rights. Boudreau (2010) provides empirical support for the positive relationship between lowered entry barriers and increasing adoption from the handheld computer industry. The author records that an increase in the number of licensed complementary hardware suppliers coincides with growing adoption of the technology platform. Despite the supportive evidence, caution is needed in assuming an exclusive causal relationship between entry barriers and platform adoption. Part of the increased adoption of the Google platforms may also be attributed to bigger popularity and thus the advantage of critical mass, superior proprietary technology in a young market or Google more actively subsidizing the supply side of the market (Katz & Shapiro, 1992; Eisenmann et al., 2009).

6.2 Development Activity and Embeddedness

Despite the higher adoption of Google platforms, the average number of applications for the Microsoft platforms is higher. Every Google Apps actor on average develops *1.36* complements and every Chrome actor develops *1.34* extensions. In contrast, the average number of complements per actor amounts *2.18* for Office365 and *1.64* for Internet Explorer. It is important to note that the recorded average for Office365 complements may be higher than the actual average, due to duplicated Microsoft Office complements in the Office365 Marketplace. Apart from higher development activity, Microsoft platform ecosystems are more interwoven and thus more robust compared to their Google peers. At present, *29.82%* of Office365 actors engage in at least one interfirm relationship with another complementor, compared to *17.14%* of the Google Apps actors. For the web browsers, these scores equal *3.18%* and *7.35%* for Google Chrome and Internet Explorer. Consequently, the average number of interfirm relationships per actor is also higher in Microsoft platform ecosystems.

The coincidence of increasing development activity and higher interconnectivity in the ecosystem gives rise to the hypothesis that increasing commitment of an actor to the ecosystem and its emergence to collaborate with other actors in the ecosystem are positively related. This hypothesis builds on the notion that power and influence are inherently relational in social and business networks alike (Hanneman & Riddle, 2005; Iansiti & Levien, 2004b). It assumes that principal actors in the ecosystem will search for interfirm relationship to ease their access to resources and means to exert power in the ecosystem. Simultaneously, smaller actors will search for technological integrations with complementary products to increase their visibility in the market or extend their potential user base.

Correlations

		Complements	Relationships
Complements	Pearson Correlation	1	,232**
	Sig. (2-tailed)		,000
	N	992	992
Relationships	Pearson Correlation	,232**	1
	Sig. (2-tailed)	,000	
	N	992	992

**. Correlation is significant at the 0.01 level (2-tailed).

FIGURE 6.1: SPSS output for Person correlation between number of complements developed and number of interfirm relationships in the Google Apps ecosystem

To examine the existence of the hypothetical relationship between development activity and embeddedness in a platform ecosystem, a Pearson correlation analysis is performed. It investigates the existence of a casual relationship between the development activity of an actor and the number of interfirm relationships in its relational portfolio. Since the average number of applications and interfirm relationships per actor for both web browser ecosystems is almost equal to 1 (Table 6.1), no significant differences exist. Accordingly, the correlation analysis is performed for the Google Apps and Office365 ecosystem that display greater interconnectivity. In the correlation analysis, the number of complements per actor is considered as the independent variable and the number of interfirm relationships initiated as the dependent variable. The result of the correlation analyses is shown in the SPSS output fragments, which can be found in Figure 6.1 for Google Apps and in Figure 6.2 for Office365.

Correlations

		Complements	Relationships
Complements	Pearson Correlation	1	,131**
	Sig. (2-tailed)		,002
	N	550	550
Relationships	Pearson Correlation	,131**	1
	Sig. (2-tailed)	,002	
	N	550	550

**. Correlation is significant at the 0.01 level (2-tailed).

FIGURE 6.2: SPSS output for Person correlation between number of complements developed and number of interfirm relationships in the Office365 ecosystem

In both ecosystems, there is a significant (*0.01* significance level) correlation between the number of complements developed by an actor and the number of interfirm relationships it initiates. For Google Apps, the obtained value equals *0.232* while for Office365 a slightly lower value is found (*0.131*). Statistical evidence thus suggests a causal relationship between increasing development activity and the embeddedness of an actor in a proprietary platform ecosystem. The positive correlation values obtained suggest that the direction of this relationship is positive. Empirical evidence thus suggests that increasing development activity coincides with increasing emergence to initiate interfirm relationships in a proprietary platform ecosystem. Consequently, the following proposition is formulated:

Proposition 2: An increase in the number of complements an actor develops will be positively related to the number of interfirm relationships it initiates in a platform ecosystem.

An explanation for the mild correlation can be sought in the number of new entrants that already initiated interfirm relationships with other actors in the Google Apps or Office365 ecosystem, in before actually joining the ecosystem. For example, a Zoho Corporation partner that enters the Google Apps ecosystem is better embedded compared to a new start-up. Other companies may first focus on enriching their product portfolio rather than an extension of their relational portfolio. An additional explanation for the mild correlation obtained from the Office365 ecosystem is the presence of a substantial amount of SharePoint developers. These developers are likely to have established interfirm relationships long before the introduction of Office365.

6.3 Partnership Model Participation

One can argue that the active partner enablement strategy of Microsoft has a profound impact on the productivity and robustness of Microsoft platform ecosystems. Microsoft can foster lock-in effects, enforce platform exclusivity for complements Eisenmann et al. (2009) and further increase (quality) control through certification by means of their partnership model (Van Angeren, Kabbedijk, et al., 2013). The Microsoft Certified Partner Network functions as an umbrella program for partnership models and certification schemes, and has a high coverage across the wider Microsoft ecosystem. In the Office365 ecosystem for instance *50.50%* of actors are certified as Microsoft partners. In contrast, the partner coverage in the Google Apps ecosystem equals *1.41%*.

Group Statistics

	PartnerStatus	N	Mean	Std. Deviation	Std. Error Mean
Complements	1,00	278	2,3129	3,35061	,20096
	,00	272	2,0625	3,38977	,20554
Relationships	1,00	278	1,1942	2,92475	,17542
	,00	272	,5221	2,50893	,15213

FIGURE 6.3: Group statistics of Microsoft partners and non Microsoft partners in the Office365 ecosystem

To examine whether partners are more active in developing complements and initiating interfirm relationships compared to non partners, an independent sample T-test is performed. Because limited partner coverage hampers the equal distribution of actors in two groups, only the Office365 ecosystem can be subjected to statistical interference. Office365 complementors are distributed in two groups based on their current partner status. Partner status is coded as a dummy variable, in which '1' corresponds to a status as Microsoft partner and '0' represents non partners. Out of the *550* complementors, *278* are acknowledged as Microsoft partners and *272* are categorized as non partners. The independent sample T-test was performed at the *95% confidence interval.* Figure 6.3 shows the SPSS output of the computed group means. A Microsoft partner in the Office365 ecosystem on average initiated *1.192* interfirm relationships with a standard deviation of *2.925*, a non partner on average had *0.5221* ties with a standard deviation equal to *2.509.* The partner category has a more skewed distribution of interfirm relationships compared to the non partners category.

Based on these figures, there is a significant difference in embeddedness between partners and non partners; $t(538) = 2.895$, $p = 0.004$ (the full SPSS output for the independent samples T-test is included in Figure 6.4). These results suggest that Microsoft partners in the Office365 ecosystem have significantly more interfirm relationships compared to non partners. Empirical evidence postulates the proposition that active partner management can foster the interconnectivity in the ecosystem.

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means	
		F	Sig.	t	df
Complements	Equal variances assumed	2,000	,158	,871	548
	Equal variances not assumed			,871	547,387
Relationships	Equal variances assumed	8,403	,004	2,890	548
	Equal variances not assumed			2,895	538,817

Independent Samples Test

		t-test for Equality of Means		
		Sig. (2-tailed)	Mean Difference	Std. Error Difference
Complements	Equal variances assumed	,384	,25045	,28741
	Equal variances not assumed	,384	,25045	,28745
Relationships	Equal variances assumed	,004	,67219	,23258
	Equal variances not assumed	,004	,67219	,23219

Independent Samples Test

		t-test for Equality of Means	
		95% Confidence Interval of the Difference	
		Lower	Upper
Complements	Equal variances assumed	-,31412	,81502
	Equal variances not assumed	-,31419	,81509
Relationships	Equal variances assumed	,21533	1,12904
	Equal variances not assumed	,21607	1,12830

FIGURE 6.4: SPSS output of independent samples T test for Microsoft partners and non Microsoft partners in the Office365 ecosystem

Proposition 3: More active partner enablement will be positively related to the network density of a platform ecosystem.

The second independent sample T-test is performed to investigate the relationship between partnership model participation and development activity. Microsoft partners on average develop *2.313* complements with a standard deviation of *3.351*, and non partners develop *2.063* complements with a standard deviation equal to *3.390*. Based on these group means, there is no significant difference in development activity between partners and non partners; $t(548) = 0.871$, $p = 0.384$. Results imply that Microsoft partners are not significantly more committed to development of complementarities compared to non partners, and that the small difference in group means may be attributed to random variation or luck. Accordingly, development scope of complementors appears unaffected by the regulations imposed around a platform.

6.4 Species in the Ecosystem

When contrasting Google Apps and Office365 with Google Chrome and Internet Explorer, comparable patterns come to surface. As shown in Table 6.1, the average number of applications developed per complementor in the cloud office suite platform ecosystems is higher than averages recorded for Google Chrome and Internet Explorer. Similarly, the initiation of interfirm relationships among Office365 and Google Apps complementors is relatively intensive compared to the incidental partnerships that can be found in the web browser ecosystems. Part of the scarcity of interfirm relationships in the Google Chrome and Internet Explorer ecosystems can be explained by the presence of numerous individuals, small companies and developer communities in the ecosystem. These type of actors will generally not actively seek for integration and influence in the ecosystem, or will not have the resources to do so. In contrast, the cloud office suite ecosystems merely harbor a handful of individual developers.

Apparent from Table 6.1 is the high connectivity between complementors in the Google Apps and Office365 ecosystems. The increased interconnectivity in turn is reflected in a higher clustering coefficient for the platform owner in these ecosystems. The cloud based office suite platforms are positioned in a specific market of enterprise applications, compared to the more generic customer market that web browsers appeal to. Complementors in the Office365 and Google Apps ecosystem appear to be predominantly organizations with a strong focus on the development of enterprise applications. The population of the web browser ecosystems seem more heterogeneous consisting of (social) media, productivity tools, web shopping and game developers that are generally unrelated. This heterogeneity is also reflected in the small and scattered cluster visualizations presented for these ecosystems in Chapter 5. Empirical evidence postulates a relationship between heterogeneity or homogeneity of the species in an ecosystem and the interconnectivity among complementors. Homogeneous populations will result in more densely connected ecosystems, since interfirm relationships are more likely to be initiated between actors that have the same specialization.

Proposition 4: The homogeneity of the species in a platform ecosystem will be positively related to its network density.

While not explicitly associated with homogeneous populations, direct complementarity is recurrently mentioned in scientific literature as a principal criterion in partner selection. Direct complementarity can be regarded from the perspective of products (Iyer et al., 2006), value networks (Varis et al., 2005) resources or capabilities (Dyer & Singh, 1998). The need for complementary resources or integrations is especially prevalent in the software industry where firms provide complex products (Basole, 2009). Iyer et al. (2006) observe differences in clustering coefficient and network density in the ecosystems of Microsoft, IBM and SAP. The IBM and Microsoft ecosystems reflect lower network density compared to the SAP ecosystem. The authors seek an explanation for the differing network topologies in the higher degree of specialization of SAP compared to Microsoft and IBM. Iyer et al. (2006) observe that the SAP ecosystem harbors most of the prominent companies in the enterprise software market and note that they actively engage into alliances among one another.

6.5 Complementor Distribution and Power Laws

The distribution of complementors based on the number of applications developed that was presented in Chapter 5, reflects commonalities for all ecosystems. The ecosystems are populated by a couple of active complementors, while the majority of complementors display limited development activity. In the Office365 ecosystem, the most active complementor developed *39* applications, which is *3.29%* of the total number of complements available for the platform. Figure 6.5 displays the histogram of the size-frequency distribution for development activity of all ecosystems, that was presented in Chapter 5. From the histograms, one can note that for the other ecosystems a similar pattern holds. The histograms display a small number of complementors that develop substantially more applications than the average, causing a long tail on the right size of the histogram. In other words, the histograms display a strong degree of right skewness.

Similar distributions have been found in a plethora of natural, social or internet phenomena that include word frequency in English language (Zipf, 1949), diameters of moon craters (Neukem & Ivanov, 1994), firm sizes in the United States (Axtell, 2001), city sizes (Newman, 2005) and web page visitor frequency (Adamic & Huberman, 2002). Characteristic for these distributions is that, when replotted with logarithmic horizontal and vertical axes, the resulting histograms follow or approach a straight line. Distributions that satisfy these conditions are said to follow a power law. To examine whether development activity in proprietary platform ecosystems potentially follows a power law,

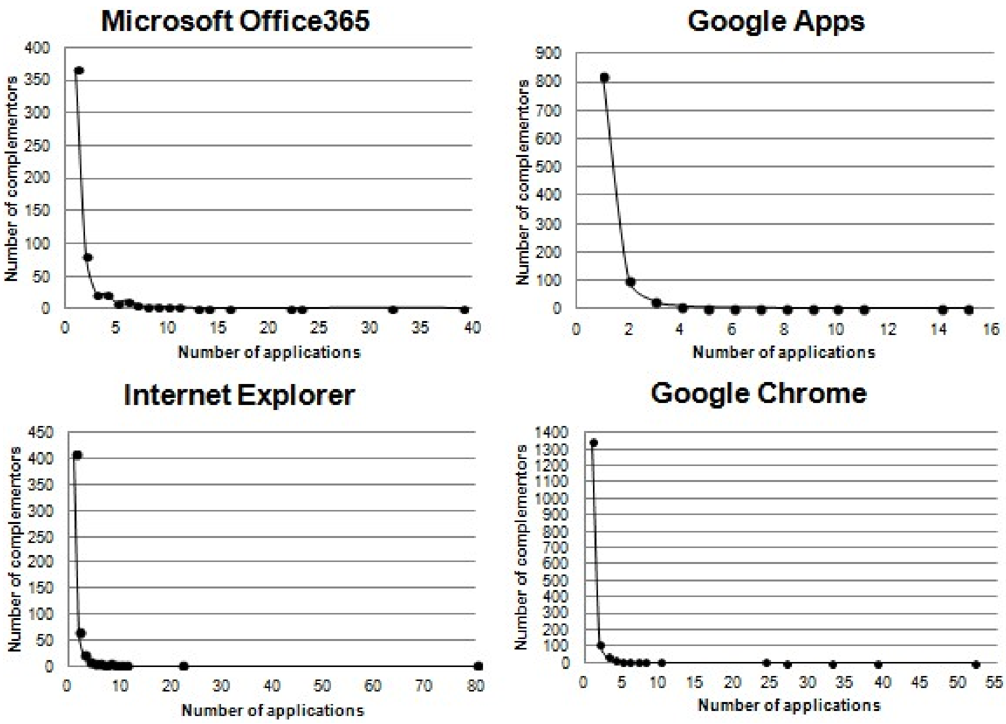

FIGURE 6.5: Histogram of size-frequency distributions for development activity

all size-frequency distributions were replotted with logarithmic axes. As shown in Figure 6.6, all four histograms closely follow a straight line, although the pattern is most apparent for Google Apps and Office365.

When y denotes the fraction of complementors that develop x or less applications, the power law for this distribution looks as shown in Equation 6.1 (Newman, 2005; Clauset, Shalizi, & Newman, 2009). In Equation 6.1, C equals e^c where c is a normalizing constant. The other parameter, α is said to be the exponent of the power law and also is a constant, usually with values $1 \leq \alpha \leq 3$. Because c is a normalizing constant, it is more usually excluded from the power law. The resulting formula is shown in Equation 6.2.

$$y = Cx^{\alpha} \tag{6.1}$$

$$y \sim x^{\alpha} \tag{6.2}$$

A special instance of a power law distribution is a Zipf distribution (Zipf, 1949). A Zipf distribution follows a power law on a rank-frequency distribution, which Zipf (1949) demonstrated in English language. The author found that "the" was the most frequently used word in the English language. Subsequent words with rank n were used with frequency $\frac{1}{n}$ of the frequency of the most popular word (Zipf, 1949). Denoted in a power

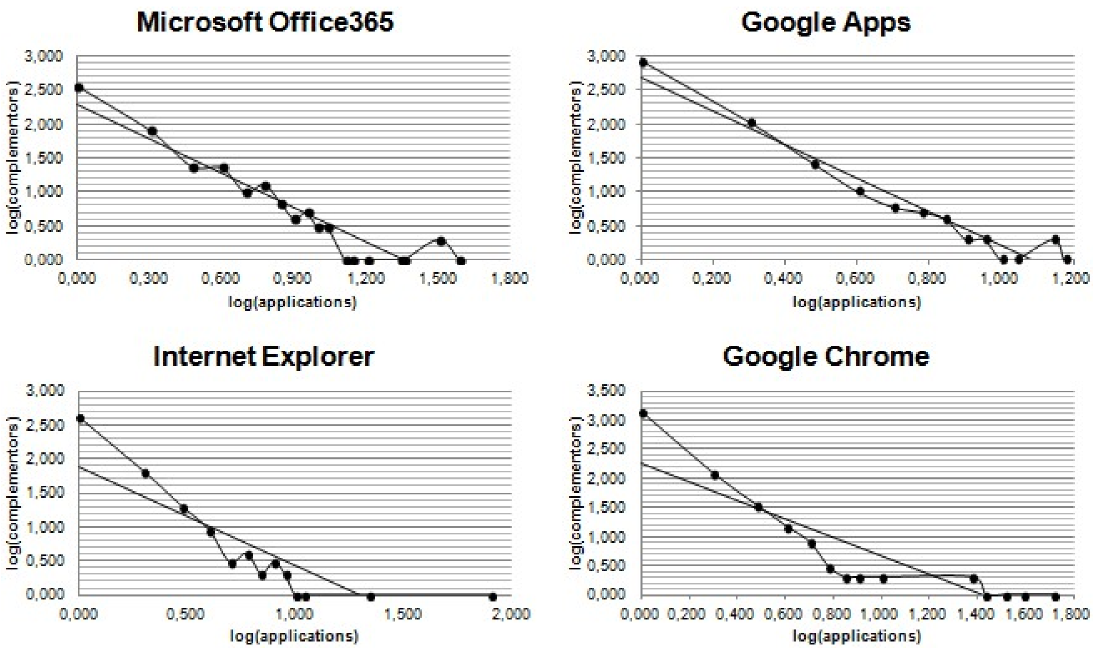

FIGURE 6.6: Log-log histogram of size-frequency distributions for development activity

law, in a Zipf distribution the value of $\alpha \sim 1$. To convert the size-frequency distributions shown in Figure 6.5 and Figure 6.6 to a rank-frequency distributions, complementors were ranked based on the number of applications they develop. For the Google Apps ecosystem, Zoho Corporation has rank $n = 1$ with *fifteen* applications, followed by Cloud Technology Solutions with rank $n = 2$ with *fourteen* applications. Due to the limited variance in ranks and applications developed per complementor, the distributions were not found to exhibit characteristics of a Zipf distribution. Accordingly, the remainder of this section only considers size-frequency distributions and their relationship to power law scaling.

A recurrent approach to testing empirical data against a hypothesized power law distribution is by means of least-squares linear regression. The power law formalized in Equation 6.2 equals the linear expression shown in Equation 6.3 (Newman, 2005; Clauset et al., 2009). The minus operator is optional, but convenient due to the negative slope of the linear line in Figure 6.6.

$$y = -\alpha log(x) + c \tag{6.3}$$

By means of least-squares linear regression, α and c can be estimated when fitting the linear function to the observed empirical data (Clauset et al., 2009). The obtained r^2 is interpreted as an indicator for the quality of the fit. Accordingly, least-squares linear regression was applied to determine the fit of a power law to the empirical data described earlier. Table 6.2 summarizes the results obtained for each platform ecosystem.

All found values fall within $1 \leq \alpha \leq 3$ and especially for the cloud office suite platforms the r^2 is remarkably high. Interpreting the results summarized in Table 6.2, statistical evidence suggests that the development activity distributions exhibit characteristics of power law scaling.

TABLE 6.2: Power law estimates and goodness of fit obtained by means of least-squares linear regression

Platform	α	**c**	$\mathbf{r^2}$
Google Apps	2.4628	2.6747	0.9478
Office365	1.6795	2.2813	0.8918
Google Chrome	1.4413	1.8818	0.7234
Internet Explorer	1.6005	2.2644	0.7911

Proposition 5: The development activity of complementors will scale in accordance with a power law.

The observed power law scaling implies that platform ecosystems are populated by a couple of large complementors, while the majority of complementors is limitedly involved in platform development. Noteworthy, the α of Google Apps is higher than that of the other three ecosystems. The development activity in the Google Apps ecosystem is more homogeneously spread over the complementors, causing variance to be lower. Accordingly, the fraction of complementors that developed x or less applications will grow faster when x increases compared to the other ecosystems. In recent work, Boudreau (2012) hypothesizes that the observed behavior contributes to a healthy and innovative ecosystem. The presence of a large number of active developers would likely imply that complementors move beyond their usual specialization, causing diseconomies of scope.

Despite the observed power law scaling, further validation is needed to confirm the adherence to the characteristics of such a distribution. In recent work, Newman (2005) and Clauset et al. (2009) examined 24 assumed power law distributions found in scientific literature. The authors found that the r^2 obtained by means of least-squares linear regression was often misleading, because the method is prone to systematic errors that can occur under normal circumstances in empirical data. Accordingly, further validation by means of the method described by Clauset et al. (2009) is necessary. Subsequently, evidenced power law scaling can be used as a starting point for growth simulation of platform ecosystems or app stores. The cumulative distribution function may for instance be used to compute the size of the ecosystem.

Chapter 7

Conclusion, Discussion and Outlook

This thesis presented results of an inductive comparative case study of proprietary platform ecosystems in the software industry, an industry that is characterized by increasing platformization, co-innovation and collaboration among stakeholders. Platform ecosystems consist of developers of complementarities to a software platform, together with the interfirm relationships among them. The research considered the cloud based office suite Google Apps and web browser Google Chrome ecosystems of Google, and the ecosystems that exist around Office365 and Internet Explorer for Microsoft. Metadata about complementors was collected by means of app store data extraction, performed with a web crawler. After that, interfirm relationships (e.g. technological partnerships, alliances) in the ecosystem were identified by creating partner lists for every complementor, based on the information available on complementor websites and CrunchBase. Thereafter, the platform ecosystems could be visualized by means of network graphs. The network graphs had a hub-and-spoke network topology, since every complementor engages in an interfirm relationship with the platform owner. All platform ecosystems exhibited similar structural properties. First, the ecosystems were sparsely connected, reflecting a limited interest in the initiation of interfirm relationships, concentration on product development or focus in relational portfolios. Second, the majority of complementors (between 66.85% and 87.58%) developed one application, while a small number of complementors develops significantly more applications than the average. Last, Each platform owner imposed some degree of entry barriers to their app store, in the form of developer fees and quality or complementary value requirements.

By means of network analysis, the research presented in this thesis uncovered the factors that shape proprietary platform ecosystems, and the extent to which these ecosystems differ across firms and platform type. The research was formulated around the following research question:

How do actors organize themselves in a proprietary platform ecosystem, and what factors influence the hereof resulting network structure?

The remainder of this chapter summarizes the main findings of the research described in this thesis, answers research questions, addresses the validation process, elaborates upon limitations, after which suggestions for future research are provided.

7.1 Findings and Implications

The research in this thesis was formulated around four sub research questions. In this section, the findings of each sub research question are presented in order to summarize the main conclusions drawn from this research.

RQ1: *How can network analysis be applied to analyze proprietary platform ecosystems?*

A literature review served as a starting point for the identification of network metrics, that describe structural properties of networks. Suitable network metrics were selected based on their applicability to analyze undirected highly centralized network graphs, and had to be optimized to calibrate large networks. Network metrics were selected to determine the following properties of an ecosystem:

- Size
- Centrality
- Density
- Sub groups
- Principal or influential actors

The applied network metrics were formally defined and mathematically formalized to foster replication. Apart from network metrics, methods were sought to compare ecosystems and permit statistical inference. However, due to scarcity of means to compare networks of different size, the contrasting of the platform ecosystems was limited to comparison of network graphs, metrics and within-case statistical inference.

RQ2: *What type of actors can be distinguished in a proprietary platform ecosystem based on their role within the network?*

The centralized hub-and-spoke network topology of proprietary platform ecosystems did not allow sufficient differentiation among actors as the platform owner is central in the ecosystem, surrounded by complementors in the periphery. To gain insight into the interactions that take place beneath the hub-and-spoke network the dataset was 'cleansed'. Platform owners and complementors that do not initiate interfirm relationships with another complementor were removed. In the description of case study results, different approaches were used to distinguish between actors. Cluster visualizations were used to distinguish central actors in influential sub groups, since these actors take principal roles in the bigger ecosystem. Similarly, actors that form relationships with members of multiple sub groups were identified as pivotal in stimulating cohesion and diffusion in the ecosystem. Network metrics such as the Bonacich power index were used to illustrate the identification of actors that are critical in retaining weakly embedded complementors. For example, both Zoho Corporation and Salesforce.com were found to initiate a large number of interfirm relationships. However, the partners of Salesforce.com were better embedded in the ecosystem compared to their Zoho Corporation counterparts. Consequently, Zoho Corporation was attributed a principal role for retention of weekly embedded complementors in the Google Apps ecosystem.

RQ3: *How do the characteristics of the firm ecosystem influence the network structure of its containing platform ecosystems?*

A review of contemporary literature on platform architecture, openness and ecosystem strategy suggested that entry barriers, ecosystem strategy and complementor retention can have a profound impact on the network structure of ecosystems. From the empirical data, coincidence of more liberal entry barriers and higher adoption rates was observed for the Google platform ecosystems, compared to their Microsoft counterparts. Despite the higher adoption rate, Microsoft ecosystems were found to be more productive and cohesive, for which subsequent analysis did provide two potential explanations. For both cloud based office suites, a correlation analysis evidenced a positive relationship between the number of complements developed by an actor and the number of interfirm relationships it initiates, which suggests that retention of complementors may lead to greater cohesion. Alternatively, the apparent difference in partnership model participation between Google (1.41%) and Microsoft (50.50%) was used as an explanation for the differences observed. An independent sample T-test compared means of partners and non partners in the Office365 ecosystem and revealed that Microsoft partners are significantly more active in initiating interfirm relationships compared to non partners. In sum, empirical evidence suggested a relationship between the characteristics of a firm and the network structure of its proprietary platform ecosystems. It implies that the platform owner can actively manipulate the network structure of its platform ecosystems.

RQ4: *What is the influence of the type of platform on the network structure of the platform ecosystem that exists around it?*

From empirical data, the cloud base office suite platforms were found to be equally cohesive, compared to incidental occurrences of relationships in the web browser ecosystems. Species analysis revealed the presence of numerous individuals and community complementors in the web browser ecosystems, whereas only a handful were found in the cloud based office suite platforms. In addition, the cloud based office suite platforms were found to be more specialized (e.g. limited to enterprise applications) compared to web browser complements that range from games to utilities and (social) media add-ons to developer tools. Empirical evidence postulated a relationship between heterogeneity or homogeneity of the species in an ecosystem and the interconnectivity among complementors. The type of platform and the type of species it attracts were thus found to have a profound impact on the network structure of the proprietary platform ecosystem.

Summarizing, proprietary platform ecosystems have been characterized as centralized hub-and-spoke networks that are sparsely connected and predominantly populated by complementors that limitedly commit to the development of complements. Further contrasting evidenced that firm characteristics, ecosystem strategy and platform type had a profound impact on the network structure. Empirical evidence and subsequent statistical inference postulated that entry barriers, partner enablement and development activity of complementors can contribute to increasing rates of adoption or increased connectivity. These findings suggest that platform owners can manipulate the network structure of their proprietary platform ecosystems by means of among others intellectual property strategy, degree of architectural control and partner enablement. However, ecosystem strategy cannot be uniformly defined for multiple platform ecosystems, due to the differing species that populate these ecosystems.

7.2 Discussion

This section presents the discussion of the findings that were summarized previously. Explication of factors such as limitations of case study research aid in placing the findings obtained by means of empirical research into perspective (Yin, 2009). In succession, results of the data collection method validation process, elaboration upon validity threats and limitations of this research are presented.

7.2.1 Validation of Data Collection Method

The research presented in this thesis used data collected from app stores, company web pages and openly accessible databases. The data collection process could be divided in three stages: (1) document analysis, (2) automated data extraction from app stores and (3) manual identification of interfirm relationships. Document analyses have been recurrently applied to study both primary and secondary data, and as such are a well established data collection method in qualitative research (Bowen, 2009).

As a consequence of the advent of mobile ecosystems and their marketplaces, app store data extraction recently emerged as a new approach to build a quantitative dataset with ready-to-use information about thousands of applications. Examples of illustrative studies on app store data extraction are the investigations by Burkard et al. (2011), Hyrynsalmi, Makila, et al. (2012) and Harman, Jia, and Zhang (2012). All investigators made use of a web crawler to retrieve data from app stores, be it or not succeeded by an interpretive or parsing step to further codify raw extracted data. The accuracy and completeness of the data collection is relatively simple to validate. The app store data extraction method is complete and accurate if all complements have been retrieved from the app store and if all required data per application are stored in the central case study database. To validate the accuracy and completeness, mere application of filters on the app store of interest sufficed. It reveals what number of applications are currently present in the app store within boundaries set. However, the actual number of unique complements may be lower, as some complements comprised part of a product line, or were listed as duplicates.

Contrary to document analysis and data extraction from app stores, manual identification of interfirm relationships is less transparent or established. In their identification of alliances and interfirm relationships, Iyer et al. (2006) make use of Lexis-Nexis, Mergent, Thomson's Corporation Securities Data Company (SDC) Platinum and International Data Corporation proprietary databases. Quaadgras (2005) also relies on the SDC Platinum database and combines it with alliance mentions in news feeds in order to reconstruct the RFID ecosystem, while Basole (2009) adds the Connexiti database as a source for the identification of interfirm relationships. While all acknowledging the reliance on proprietary sources, none of the authors has validated the data collection method to the extent at which statements can be made about its accuracy and completeness. Accordingly, the validation of the data collection method used in this thesis concentrated on the manual reconstruction of interfirm relationships. In particular, the question was whether company web pages and CrunchBase were reliable as sources to reconstruct a complete and accurate overview of interfirm relationships.

After the completion of the data collection and analysis for the cloud office suite platforms, *35* Google Apps complementors were contacted by e-mail (as retrieved during the app store data extraction). Complementors were selected based on their development activity and relationship portfolio. The *eighteen* complementors with the most developed applications and the *seventeen* complementors with the most interfirm relationships were contacted. Each complementor received an e-mail that contained its portfolio of interfirm relationships and the cluster visualization of Google Apps that was presented in Chapter 5. Under circumstance of strict anonymity, complementors were asked to verify the accuracy and completeness of the interfirm relationship portfolio that was compiled for them. In return for their participation, complementors received a comparison report that describes the Google Apps ecosystem and compares it to the Office365 ecosystem. The report is included in Appendix B.

In total, *ten (29%)* Google Apps complementors responded to the small sample e-mail survey. Nine respondents indicated that the compiled interfirm relationship portfolio for their company was accurate and complete. None of the respondents indicated that companies were wrongfully added to their relationship portfolio. One responded indicated that the portfolio compiled was incomplete. The respondent, however, indicated to *"deliberately not publish certain partnerships"* emphasizing the drawbacks of reliance on proprietary sources. The results of the validation suggest that the data collection method is effective in capturing interfirm relationships. Nevertheless, further validation is needed to make statements about the exact level of accuracy of the method.

7.2.2 Validity Threats and Limitations

Despite the positive results obtained during the validation, like any other exploratory research this study has a number of limitations. In order to systematically create an extensive overview of four proprietary platform ecosystems, design choices were made to keep the data collection demands manageable and provide standardization by means of a uniformly applied case study protocol. Each platform ecosystem was created by collecting interfirm relationships from company web pages, news feeds and CrunchBase, a tedious task. While a preliminary validation provides strong indications that these sources provide an extensive overview, the addition of sources or the use of different data collection strategies should bring further insight on the exact accuracy. The use of proprietary databases such as SDC Platinum proved effective on an industry scale, however, it remains unclear if an alliance perspective alone provides sufficient depth for the analysis of platform ecosystems. Certainly since the granularity in these ecosystems is finer.

Inherent in the problem domain is the strong reliance on proprietary sources when visualizing platform ecosystems. The lack of transparency becomes evident when complementors indicate to engage in technological partnerships or to employ partnership models, but omit to list their actual partners. Accordingly, the partner activity of such a company is neglected, while in practice it might be involved in numerous interfirm relationships. To limit the effect, CrunchBase was added as an additional data source, and interfirm relationship mentions were treated symmetric to increase coverage. In addition, at present it remains unclear how many of the identified interfirm relationships can be directly attributed to the platform. At least part of the identified interfirm relationships are likely to reflect industry-wide ties, rather than platform-specific collaborations. In-depth case studies with, or surveys among complementors should provide insight into the degree of interfirm relationships that can be directly related to a platform.

The data collection for each individual case study proceeded in accordance with a predefined case study protocol to benefit direct unbiased comparison. However, the lack of statistical approaches to compare multiple networks of different size, species and type draw limitations upon the depth of analyses. Consequently, the contrasting of proprietary platform ecosystems had to be limited to visual inspection and metrics comparison, complemented by within-case statistical inference. As such, the research presented in this thesis only partly succeeded in following an inductive case study approach (Eisenhardt, 1989). Observations were formulated into propositions, each of which to be addressed in future research to establish theories. An example is the observed characteristics of power law scaling in complementor development activity, that need further validation by means of the method described by Clauset et al. (2009).

7.3 Future Research

The research results presented in this thesis provide a step towards better understanding of dynamics and forces at play in proprietary platform ecosystems. The results presented give rise to potential avenues for future research, which are laid out in the remainder of this section.

7.3.1 Niche Identification

The scope of the research project presented in this thesis emphasizes software products as complementarities to a software platform. However, in segments of the software industry such as the enterprise software market, service providers play an equally important role (Cusumano, 2007; Popp & Meyer, 2010). Services can be offered as complements to the software platform, which is the case for implementation, training or consultancy services. Services are usually listed in the marketplace of the platform. Apart from offering direct service complements to the software platform, service providers can act as brokers between developers of multiple software complements, or between software complementor and end-user. The service provider can, for example, integrate two software complements, an indirect relationship between two software complements that is not visible with a scope that solely considers software products.

Extending the scope of the platform ecosystem beyond software complements can thus aid in providing a more complete picture of the ecosystem. In addition, the inclusion of service providers can be used as a starting point for 'niche identification' in the ecosystem. Network analysis provides means for so called 'structural hole analysis' (Hanneman & Riddle, 2005), to identify weak spots in the ecosystem. Based on interconnectivity in the ecosystem and node properties (i.e. metadata about complementors), opportunities can be identified as they arise. Longitudinal case studies that incorporate evaluation cycles could provide a principal avenue towards accurate and dynamic identification of niches.

7.3.2 Effect of Multiplex Relationships in Ecosystems

The classification of relationships as being present or absent is a simplification of reality. The software industry is characterized by intense collaboration and competition, recurrently occurring at the same time (Bengtsson & Kock, 2000; Iyer et al., 2006; Patrucco, 2012; Rosenkopf & Schilling, 2007). Two companies may at a given point in time have initiated a technological partnership and a distribution partnership, while also being competitors. The result of this complex relational schema is a web of software companies that are interwoven by means of a plethora of interfirm relationships, each with their own weight. So far this aspect of the software industry remained unaddressed, providing limited means for understanding behavior of actors in software ecosystems.

A potential way to address this more realistic perspective on the software industry is through the reconstruction of multiplex networks. In multiplex networks ties between nodes are valued based on their strength, and two nodes may be connected more than once if multiple interfirm relationships exist. This multiplex network can be decomposed to investigate how future relationship initiation relates to already existing collaborations. Subsequently, this also enables the possibility for statistical inference, since the body of knowledge on sampling in network analysis is well established. Recent examples of multiplex network analyses include the study of multiplex communication networks between employees by Lee and Monge (2011), and Powell, White, Koput, and Owen-Smith (2005) who study the multiplex network in the biotechnology industry that constitutes interfirm research, finance, licensing intellectual property and marketing ties.

7.3.3 Automated Collection of Interfirm Relationships

The identification of interfirm relationships is a tedious task. It involves thorough inspection of company websites and querying databases, and as such is difficult to apply to larger samples, or on a recurrent basis. To benefit replication, and the applicability of the demonstrated ecosystem analyses by platform owners, automation of the identification of interfirm relationships is a necessity. The automation, however, is complex due to a lack of standardization in company websites and the way in which they enumerate partners. Nevertheless, the software engineering research community is well equipped to provide means for collecting and parsing non standardized data.

7.3.4 Conduct Surveys among Complementors

The research that was presented in this thesis produced propositions about observed factors that influence the network structure of proprietary platform ecosystems. Herein the perspective of the actors that develop complements to the software platform is deficient. Future research should produce surveys among complementors in platform ecosystems to address the decisions that are made by these complementors. Potential research directions include examining the influence of platform architecture, supply side subsidies or entry barriers on platform selection and attaining insight into the extent to which interfirm relationships can be directly related to the platform of interest.

7.3.5 Longitudinal Studies of Ecosystem Evolution

A static analysis such as the one presented in this thesis considers an ecosystem at a single point in its evolution cycle. However, networks tend to change over time. This is especially the case for the software industry, where fast-paced innovation and technological change are prevalent. Consequently, actors enter, depart, or form interfirm relationships in the ecosystem on an ongoing basis. The findings presented in this thesis merely provide a first step towards better understanding of proprietary platform ecosystems. Therefore, a longitudinal case study of proprietary platform ecosystems could benefit a wealth to the understanding of the fast-paced dynamics characteristic for the software industry. At the same time, longitudinal studies could provide further insight into the power law scaling observed in the size-frequency distribution for development activity. Subsequently, these observations could be further formalized to work towards models for longitudinal ecosystem simulation and prediction. The data and research presented in this thesis can function as a blueprint for future studies of the same or different ecosystems. To stimulate replication, datasets will be made available upon request.

References

Adamic, L., & Huberman, B. (2002). Zipf's law and the internet. *Glottometrics*, *3*, 143-150.

Alspaugh, T., Asuncion, H., & Scacchi, W. (2009). The Role of Software Licenses in Open Architecture Ecosystems. *Proceedings of the First International Workshop on Software Ecosystems*, 4-18.

Anderson, C., Wasserman, S., & Crouch, B. (1999). A p* primer: logit models for social networks. *Social Networks*, *21*(1), 37-66.

Anvaari, M., & Jansen, S. (2010). Evaluation Architectural Openness in Mobile Software Platforms. *Proceedings of the Second International Workshop on Software Ecosystems*, 85-92.

Arksey, H., & O'Malley, L. (2005). Scoping studies: towards a methodological framework. *International Journal of Social Research Methodology*, *8*(1), 19-32.

Armstrong, M. (2006). Competition in two-sided markets. *RAND Journal of Economics*, *37*(3), 668-691.

Axtell, R. (2001). Zipf Distribution of U.S. Firm Sizes. *Science*, *293*(5536), 1818-1820.

Baldwin, C., & Woodard, C. (2009). The architecture of platforms: A unified view. In A. Gawer (Ed.), *Platforms, markets and innovation* (p. 19-45). Cheltenham, UK: Edward Elgar Publishing.

Barbosa, O., & Alves, C. (2011). A Systematic Mapping Study on Software Ecosystems. *Proceedings of the Third International Workshop on Software Ecosystems*, 15-26.

Basole, R. (2009). Visualization of interfirm relations in a converging mobile ecosystem. *Journal of Information Technology*, *24*, 144-159.

Bastian, M., Heymann, S., & Jacomy, M. (2009). Gephi: An Open Source Software for Exploring and Manipulating Networks. *Proceedings of the Third International Conference on Weblogs and Social Media*.

Bengtsson, M., & Kock, S. (2000). 'Coopetition' in Business Networks – to Cooperate and Compete Simultaneously. *Industrial Marketing Management*, *29*(5), 411-426.

Blondel, V., Guillaume, J., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics*, *5*(10).

Bollobas, B. (1998). *Modern graph theory.* USA: Springer.

Bonacich, P. (1987). Power and Centrality: A Family of Measures. *American Journal of Sociology*, *92*(5), 1170-1182.

Bonacich, P. (2007). Some unique properties of eigenvector centrality. *Social Networks*, *29*(4), 555-564.

Bosch, J. (2002). Maturity and Evolution in Software Product Lines: Approaches, Artefacts and Organization. *Proceedings of the Second International Conference on Software Product Lines*, 257-271.

Bosch, J. (2009). From software product lines to software ecosystems. *Proceedings of the 13th International Software Product Line Conference*, 1-10.

Boudreau, K. (2010). Open Platform Strategies and Innovation: Granting Access vs. Devolving Control. *Management Science*, *56*(10), 1849-1872.

Boudreau, K. (2012). Let a Thousand Flowers Bloom? An Early Look at Large Numbers of Software App Developers and Patterns of Innovation. *Organization Science*, *23*(5), 1409-1427.

Bowen, G. (2009). Document analysis as a qualitative research method. *Qualitative Research Journal*, *9*(2), 27-40.

Brereton, P., Kitchenham, B., Budgen, D., & Li, Z. (2008). Using a protocol template for case study planning. *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, 41-48.

Bresnahan, T., & Greenstein, S. (1999). Technological competition and structure of the computer industry. *The Journal of Industrial Economics*, *47*(1), 1-40.

Brinkkemper, S. (1996). Method engineering: engineering of information systems development methods and tools. *Information and Softwre Technology*, *38*(4), 275-280.

Browning, T. (2001). Applying the Design Sutrcture Matrix to System Decomposition and Integration Problems: A Review and New Directions. *IEEE Transactions on Engineering Management*, *48*(3), 292-306.

Burkard, C., Draisbach, T., Widjaja, T., & Buxmann, P. (2011). Software ecosystems: Vendor-sided characteristics of online marketplaces. *Proceedings of the 41st Annual Conference of the Gesellschaft fur Informatik*.

Burt, R., & Minor, M. (1983). *Applied network analysis: A methodological introduction.* Beverly Hills, USA: Sage Publications, Inc.

Butler, M. (2011). Android: Changing the Mobile Landscape. *IEEE Pervasive Computing*, *10*(1), 4-7.

Clauset, A., Shalizi, C., & Newman, M. (2009). Power-Law Distribution in Empirical Data. *Society for Industrial and Applied Mathematics Review*, *51*(4), 661-703.

Cusumano, M. (2007). The changing labyrinth of software pricing. *Communications of the ACM*, *50*(7), 19-22.

Cusumano, M. (2010a). *Staying power: Six enduring principles for managing strategy and innovation in an uncertain world.* New York, NY, USA: Oxford University Press.

Cusumano, M. (2010b). Technology strategy and management: The evolution of platform thinking. *Communications of the ACM*, *53*(1), 32-34.

Cusumano, M. (2012). Platforms versus products: Observations from the literature and history. *Advances in Strategic Management*, *29*, 35-67.

Dellinger, A., & Leech, N. (2007). Toward a unified validation framework in mixed methods research. *Journal of Mixed Methods Research*, *1*(4), 309-332.

Den Hartigh, E., Tol, M., & Visscher, W. (2006). The health measurement of a business ecosystem. *Proceedings of the European Network on Chaos and Complexity Research and Management Practice Meeting*.

Dyer, J., & Singh, H. (1998). The Relational View: Cooperative strategy and sources of interorganizational competitive strategy. *Academy of Management Review*, *23*(4), 660-679.

Economides, N., & Katsamakas, E. (2006). Two-Sided Competition of Proprietary vs. Open Source Technology Platforms and the Implications for the Software Industry. *Management Science*, *52*(7), 1057-1071.

Eisenhardt, K. (1989). Building Theories from Case Study Research. *The Academy of Management Review*, *14*(4), 532-550.

Eisenhardt, K., & Graebner, M. (2007). Theory building from cases: Opportunities and challenges. *Academy of Management Journal*, *50*(1), 25-32.

Eisenmann, T., Parker, G., & Van Alstyne, M. (2006). Strategies for Two-Sided Markets. *Harvard Business Review*, *84*(10), 92-101.

Eisenmann, T., Parker, G., & Van Alstyne, M. (2009). Opening platforms: How, when and why? In A. Gawer (Ed.), *Platforms, markets and innovation* (p. 131-162). Cheltenham, UK: Edward Elgar Publishing.

Eisenmann, T., Parker, G., & Van Alstyne, M. (2011). Platform Envelopment. *Strategic Management Journal*, *32*(12), 1270-1285.

Farrell, J., & Saloner, G. (1985). Standardization, compatibility, and innovation. *The RAND Journal of Economics*, *16*(1), 70-83.

Faust, K., & Skvoretz, J. (2002). Comparing networks across space and time, size and species. *Sociological Methodology*, *32*(1), 267-299.

Ferguson, C., & Morris, C. (2003). *Computer wars: The post-ibm world.* Washington, DC, USA: Beard Books.

Festiger, L. (1949). The analysis of sociograms using matrix algebra. *Human Relations*, *2*, 153-158.

Fransman, M. (2002). Mapping the evolving telecoms industry: uses and shortcomings of the layer model. *Telecommunications Policy*, *26*(10), 473-483.

Freeman, L. (1979). Centrality in social networks conceptual clarification. *Social Networks*, *1*(3), 215-239.

Fricker, S. (2009). Specification and Analysis of Requirements Negotiation in Software Ecosystems. *Proceedings of the First International Workshop on Software Ecosystems*, 19-33.

Gao, L., & Iyer, B. (2006). Analyzing complmentarities using software stacks for software industry acquisitions. *Journal of Management Information Systems*, *23*(2), 119-147.

Garton, L., Haythornthwalte, C., & Wellman, B. (1997). Studying Online Social Networks. *Journal of Computer-Mediated Communication*, *3*(1).

Gawer, A. (2009a). Platform dynamics and strategies: From products to services. In A. Gawer (Ed.), *Platforms, markets and innovation* (p. 45-76). Cheltenham, UK: Edward Elgar Publishing.

Gawer, A. (2009b). Platforms, markets and innovation: An introduction. In A. Gawer (Ed.), *Platforms, markets and innovation* (p. 1-16). Cheltenham, UK: Edward Elgar Publishing.

Gawer, A., & Cusumano, M. (2002). *Platform leadership: How intel, microsoft, and cisco drive industry innovation.* Boston, MA, USA: Harvard Business School Press.

Gawer, A., & Cusumano, M. (2008). How companies become platform leaders. *MIT Sloan Management Review*, *49*(2), 28-35.

Gawer, A., & Henderson, R. (2007). Platform owner entry and innovation in complementary markets: Evicdence from Intel. *Journal of Economics and Management Strategy*, *16*(1), 1-34.

Golumbic, M. (2004). *Algorithmic graph theory and perfect graphs* (2nd ed.). Hungary: Elsevier.

Granovetter, M. (1976). Network sampling: Some first steps. *The American Journal of Sociology*, *81*(6), 1287-1303.

Hagel III, J., Brown, J., & Davidson, L. (2008). Shaping strategy in a world of constant disruption. *Harvard Business Review*, *86*(10), 80-89.

Håkansson, H., & Snehota, I. (1995). *Developing relationships in business networks.* London, UK: Routledge.

Hammel, T., Phelps, T., & Kuettner, D. (2002). The re-engineering of Hewlett-Packards CD-RW supply chain. *Supply Chain Management: An International Journal*, *7*(3), 113-118.

Hanneman, R., & Riddle, M. (2005). *Introduction to social network methods.* Riverside, CA: University of California.

Hanssen, G., & Dyba, T. (2012). Theoretical foundations of software ecosystems. *Proceedings of the Fourth International Workshop on Software Ecosystems*, 6-17.

Harman, M., Jia, Y., & Zhang, Y. (2012). App Store Mining and Analysis: MSR for App Stores. *Proceedings of the Nineth Working Conference on Mining Software Repositories*, 108-111.

Hearn, G., & Pace, C. (2005). Value-creating ecologies: understanding next generation business systems. *Foresight*, *8*(1), 55-65.

Host, M., & Runeson, P. (2007). Checklists for Software Engineering Case Study Research. *Proceedings of the First International Symposium on Empirical Software Engineering and Management*, 479-481.

Huang, P., Caccagnoli, M., & Forman, C. (2009). When do ISVs join a Platform Ecosystem? Evidence from the Enterprise Software Industry. *Proceedings of the International Conference on Information Systems*.

Hyrynsalmi, S., Makila, T., Jarvi, A., A. Suominen, Seppanen, M., & Knuutila, T. (2012). App Store, Marketplace, Play! An Analysis of Multi-Homing in Mobile Software Ecosystems. *Proceedings of the Fourth International Workshop on Software Ecosystems*, 59-72.

Hyrynsalmi, S., Suominen, A., Makila, T., Jarvi, A., & Knuutila, T. (2012). Revenue Models of Application Developers in Android Market Ecosystem. *Proceedings of the Third International Conference on Software Business*, 209-222.

Iansiti, M., & Levien, R. (2004a). *The keystone advantage: What the new dynamics of business ecosystems mean for strategy, innovation, and sustainability.* Boston, MA, USA: Harvard Business School Press.

Iansiti, M., & Levien, R. (2004b). Strategy as Ecology. *Harvard Business Review*, *82(3)*, 68-78.

Iyer, B. (2012). EcoSysNetworks: A Method for Visualizing Software Ecosystems. *Proceedings of the Fourth International Workshop on Software Ecosystems*, 1-5.

Iyer, B., Lee, C., & Venkatraman, N. (2006). Managing in a Small World Ecosystem: Some Lessons from the Software Sector. *California Management Review*, *48*(3), 28-47.

Jansen, S., Brinkkemper, S., & Finkelstein, A. (2007). Providing Transparency In The Business Of Software: A Modeling Technique For Software Supply Networks. *Proceedings of the 8th Working Conference on Virtual Enterprises*, 677-686.

Jansen, S., Brinkkemper, S., & Finkelstein, A. (2009). Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems. *Proceedings of the First International Workshop on Software Ecosystems*, 34-48.

Jansen, S., Brinkkemper, S., Hunink, I., & Demir, S. (2008). Pragmatic and Opportunistic Reuse in Innovative Start-up Companies. *IEEE Software*, *25*(6), 42-46.

Jansen, S., Brinkkemper, S., Souer, J., & Luinenburg, L. (2012). Shades of gray: Opening up a software producing organization with the open software enterprise model. *The Journal of Systems and Software*, *85*(7), 1495-1510.

Jansen, S., & Cusumano, M. (2012). Defining Software Ecosystems: A Survey of Software Platforms and Business Network Governance. *Proceedings of the Fourth International Workshop on Software Ecosystems*, 40-58.

Jansen, S., Finkelstein, A., & Brinkkemper, S. (2009). A sense of community: A research agenda for software ecosystems. *Proceedings of the 31st ICSE Conference on Software Engineering, New and Emerging Research Track*, 87-90.

Kabbedijk, J., & Jansen, S. (2011). Steering Insight: An Exploration of the Ruby Software Ecosystem. *Proceedings of the Second International Conference on Software Business*, 44-55.

Katz, M., & Shapiro, C. (1992). Product introduction with network externalities. *Journal of Industrial Economics*, *40*(2), 55-83.

Kerlinger, F. (1986). *Foundations of behavioral research.* Net York, NY, USA: Holt, Rinehart and Winston.

Kitchenham, B., & Pickard, L. (1998). Evluating Software Engineering Methods and Tools, Part 9: Quantitative Case Study Methodology. *ACM Software Engineering Notes*, *23*(1), 24-26.

Kittlaus, H., & Clough, P. (2009). *Software product management and pricing: Key success factors for software organizations.* Heidelberg, Germany: Springer.

Krishan, V., & Gupta, G. (2001). Appropriateness and impact of platform-based product development. *Management Science*, *47*(1), 52-68.

LaMartina, M., Cai, Y., MacCormack, A., & Rusnak, J. (2008). Analyzing the Evolution of Large-Scale Software Systems using Design Structure Matrices and Rule Therory: Two Exploratory Cases. *Proceedings of the Seventh Working Conference on Software Architecture*, 83-92.

Lavie, D. (2007). Alliance portfolios and firm performance: A study of value creation and appropriation in the u.s. software industry. *Strategic Management Journal*(12), 1187-1212.

Lee, S., & Monge, P. (2011). The coevolution of multiplex communication networks in organizational communities. *Journal of Communication*, *61*, 758-779.

Madey, G., Freeh, V., & Tynan, R. (2004). Modeling the Free/Open Source Software Community: A Quantitative Approach. In S. Koch (Ed.), *Free/Open Source Software Development* (p. 203-220). Hershey, PA, USA: Idea Group Publishing.

Manikas, K., & Hansen, K. (2012). Software ecosystems: A systematic literature review. *Journal of Systems and Software*, *86*(5), 1294-1306.

Martin, J. (1999). A General Permutation-Based QAP Analysis Approach for Dyadic Data from Multiple Groups. *Connections*, *22*(2), 50-60.

Martin, M., & Ishii, K. (2002). Design for variety: developing standardized and modularized product platform architectures. *Research Engineering Design*, *13*(4), 213-235.

McGrath, M. (2000). *Product strategy for high technology companies* (2nd ed.). New York, NY, USA: McGraw-Hill.

Mens, T., & Goeminne, M. (2011). Analysing the evolution of social aspects of open source software ecosystems. *Proceedings of the Third International Workshop on Software Ecosystems*, 1-14.

Messerschmitt, D., & Szyperski, C. (2005). *Software ecosystem: Understanding an indispensable technology and industry.* MIT Press.

Meyer, M., & Lehnerd, A. (1997). *The power of product platforms: Building value and cost leadership.* Net York, USA: Free Press.

Moore, J. (1993). Predators and Prey: A New Ecology of Competition. *Harvard Business Review*, *71*(3), 75-83.

Moore, J. (1996). *The death of competition: Leadership and strategy in the age of business ecosystems.* New York, NY, USA: Harper Business.

Muffato, M., & Roveda, M. (2002). Product architecture and platforms: a conceptual framework. *International Journal of Technology Management*, *24*(1), 1-16.

Muller, R., Kiji, B., & Martens, J. (2011). A Comparison of Inter-Organizational Business Models of Mobile App Stores: There is more than Open vs. Closed. *Journal of Theoretical and Applied Electronic Commerce Research*, *6*(2), 63-76.

Neukem, C., & Ivanov, B. (1994). Crater size distributions and impact probabilities on earth from lunar, terrestialplanet, and asteroid cratering data. In T. Gehrels (Ed.), *Hazards Due to Comets and Asteroids* (p. 359-416). Tucson, AZ, USA: University of Arizona Press.

Newman, M. (2005). Power laws, Pareto Distributions and Zipf's law. *Contemporary Physics*, *46*(5), 323-351.

Parker, G., & Van Alstyne, M. (2005). Two-Sided Network Effects: A Theory of Information Product Design. *Management Science*, *51*(10), 1494-1504.

Patrucco, P. (2012). Innovation platforms, complexity and the knowledge-intensive firm. In M. Dietrich & J. Krafft (Eds.), *Handbook on the economics and theory of the firm* (p. 358-374). Cheltenham, UK: Edward Elgar Publishing.

Pattison, P., & Wasserman, S. (1999). Logit models and logic regressions for social networks: II. Multivariate relations. *British Journal of Mathematical and Statistical Psychology*, *52*(2), 169-193.

Peltoniemi, M., & Vuori, E. (2004). Business ecosystem as the new approach to complex adaptive business environments. *Proceedings of the Fourth Annual eBusiness Research Forum*, 267-281.

Perrons, R. (2009). The open kimono: How Intel balances trust and power to maintain platform leadership. *Research Policy*, *38*(8), 1300-1312.

Pine, B. (1993). *Mass customization: The new frontier in business competition.* Boston, MA, USA: Harvard Business School Press.

Popp, K., & Meyer, R. (2010). *Profit from software ecosystems: Professional edition, business models, ecosystems and partnerships in the software industrys* (none, Ed.). Books on Demand GmbH.

Powell, W., White, D., Koput, K., & Owen-Smith, J. (2005). Network Dynamics and Field Evolution: The Growth of Interorganizational Collaboration in the Life Sciences. *American Journal of Sociology*, *110*(4), 1132-1205.

Pressman, R. (2004). *Software engineering: A practitionerś approach* (6th ed.). McGraw Hill Higher Education.

Quaadgras, A. (2005). Who joins the platform? the case of the RFID business ecosystem. *Proceedings of the 38th Hawaii International Conference on System Sciences*, 269-278.

Robertson, D., & Ulrich, K. (1998). Planning for Product Platforms. *Sloan Management Review*, *39*(4), 19-31.

Robins, G., Pattison, P., Kalish, Y., & Lusher, D. (2007). An introduction to random graph (p*) models for social networks. *Social Networks*, *29*(2), 173-191.

Rochet, J., & Tirole, J. (2003). Platform Competition in Two-Sided Markets. *Journal off the European Economic Assosication*, *1*(4), 990-1029.

Rosenkopf, L., & Schilling, M. (2007). Comparing alliance network structure across industries: Observations and explanations. *Strategic Enterpreneurship Journal*, *1*(3-4), 191-209.

Runeson, P., & Host, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, *14*, 131-164.

Rysman, M. (2009). The Economics of Two-Sided Markets. *Journal of Economic Perspectives*, *23*(3), 125-143.

Sanchez, R., & Mahoney, J. (1997). Modularity, Flexibility and Knowledge Management in Product and Organizational Design. *IEEE Engineering Management Review*, *25*(4), 50-61.

Sanderson, S., & Uzumeri, M. (1995). Managing product families: The case of the Sony Walkman. *Research Policy*, *24*(5), 761-782.

Santamaria, L., & Surroca, J. (2011). Matching the Goals and Impacts of R&D Collaboration. *European Management Review*, *8*(2), 95-109.

Schilling, M. (2009). Protecting or diffusing a technology platform: Tradeoffs in appropriability, network externalities and architectural control. In A. Gawer (Ed.), *Platforms, markets and innovation* (p. 192-218). Cheltenham, UK: Edward Elgar Publishing.

Scott, J. (2000). *Social network analysis: A handbook* (2nd ed.). Gateshead, UK: Sage Publications, Inc.

Suh, E., De Weck, O., & Chang, D. (2007). Flexible product platforms: framework and case study. *Research Engineering Design*, *18*(2), 67-89.

Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research Policy*, *24*(3), 419-440.

Van Angeren, J., Blijleven, V., Jansen, S., & Brinkkemper, S. (2013). Interfirm Relationships in Software Ecosystems: Implications for Business Strategy and Software Architecture. *In Preparation*.

Van Angeren, J., Kabbedijk, J., Jansen, S., & Popp, K. (2011). A Survey of Associate Models used within Large Software Ecosystems. *Proceedings of the Third International Workshop on Software Ecosystems*, 27-39.

Van Angeren, J., Kabbedijk, J., Popp, K., & Jansen, S. (2013). Managing software ecosystems through partnering. In M. Jansen S. Cusumano & S. Brinkkemper (Eds.), *Software ecosystems: Analyzing and managing business networks in the software industry* (p. 85-102). Cheltenham, UK: Edward Elgar Publishing.

Van de Weerd, I., & Brinkkemper, S. (2008). Meta-Modeling for Situational Analysis and Design Methods. In M. Syed & S. Syed (Eds.), *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications* (p. 38-58). Hershey: Idea Group Publishing.

Varis, J., Kuivalainen, O., & Saarenketo, S. (2005). Partner Selection for International Marketing and Distribution in Corporate New Ventures. *Journal of International Entrepreneurship*, *3*(1), 19-36.

Venkatraman, N., Lee, C., & Iyer, B. (2008). Interconnect to win: The joint effects of business strategy and network positions on the performance of software firms. In J. Baum & T. Rowley (Eds.), *Network Strategy* (p. 391-425). Bingley, UK: JAI Press.

Wasserman, A. (2010). Software Engineering Issues for Mobile Application Development. *Proceedings of the Workshop on the Future of Software Engineering Research*.

Watts, D., & Strogatz, S. (1998). Collective dynamics of small world networks. *Nature*, *393*, 440-442.

West, J. (2003). How Open is Open Enough? Melding Proprietary and Open Source Platform Strategies. *Research Policy*, *32*(7), 1259-1285.

Wheelwright, S., & Clark, K. (1992). Creating Project Plans to Focus Software Development. *Harvard Business Review*, *70*(2), 67-83.

Xu, L., & Brinkkemper, S. (2007). Concepts of Product Software. *European Journal of Information Systems*, *16*, 531-541.

Yin, R. K. (2009). *Case study research: Design and methods.* Thousand Oaks, CA, USA: Sage Publications, Inc.

Zhu, F., & Iansiti, M. (2012). Entry into platform-based markets. *Strategic Ma*, *33*(1), 88-106.

Zipf, G. (1949). *Human behaviour and the principle of least effort.* Reading, MA, USA: Addison-Wesley.

Zirpoli, F., & Caputo, M. (2002). The nature of buyer-supplier relationships in co-design activities: The Italian auto industry case. *Journal of Operation and Production Management*, *22*(12), 1389-1410.

Appendix A

Case Study Protocol

A.1 Background

This research project addresses the network structure of platform ecosystems and the factors that shape networks of complementors. The software industry becomes increasingly interconnected and interdependent. The success of a software platform depends on the number of other organizations that surround it. The success and value of Android, for example, increases as the number of mobile applications (i.e. complements) for Android increases, while the value of an ERP platform is determined by number of partners that build domain-specific extensions. As a consequence of this increasing need for relationships, networks of interrelated actors form around organizations and platforms, that are referred to as software ecosystems. A software ecosystem is defined as "a set of businesses functioning as a unit and interacting with a shared market for software and services, together with the relationships among them" (Jansen, Finkelstein, & Brinkkemper, 2009, p. 1). A platform ecosystem forms itself around a software platform, such as Microsoft Windows, Google Apps or Facebook and is constituted of all complementors and the relationships they establish among each other.

While different researchers emphasize the importance of co-innovation and collaboration among different complementors in the ecosystem (Gawer & Cusumano, 2002; Baldwin & Woodard, 2009), so far little is known about the way in which complementors organize themselves around a software platform. As a consequence, platform owners have limited insight in the mechanisms at play in their platform ecosystems and therefore lack guidance on how to stimulate collaboration in their platform ecosystems. To address this deficiency this research project aims to provide an answer to the following research question: **"How do actors organize themselves in a proprietary platform ecosystem, and what factors influence the hereof resulting network**

structure?". Additionally, the problem domain is broken down in the following sub research questions, each of which is regarded as an elementary step in answering the main research question.

1. How can network analysis be applied to analyze proprietary platform ecosystems?
2. What type of actors can be distinguished in a proprietary platform ecosystem based on their role within the network?
3. How do the characteristics of the firm ecosystem influence the network structure of its containing platform ecosystems?
4. What is the influence of the type of platform on the network structure of the platform ecosystem that exists around it?

A.2 Design

The subject of study in this case study research project will be the actors that offer complements to a software platform (e.g. applications, plug-ins or other extensions), together with the relationships among them. These complements are distributed to the end user by means of extension markets (Jansen & Cusumano, 2012), such as application stores or plug-in listings, that usually are maintained, managed and governed by the owner of the software platform. The Apple AppStore is an example of an extension market, in which complementors offer applications for the iOS mobile operating system.

To provide an answer to the (sub) research questions that shape this research project, a multiple case design is chosen (Eisenhardt, 1989; Yin, 2009). Four separate case studies are conducted – each in adherence to the research procedures as described in this case study protocol – to provide for increased comparability of the respective case study subjects (i.e. complementor ecosystems). As complementor ecosystems can be compared by means of different approaches and analysis techniques, an embedded case study approach is preferred. Subsequently, the composition of the complementor ecosystem (e.g. the types of actors that together constitute the ecosystem) and the network structure (e.g. the characteristics of the ecosystem as a function of the symmetric relationships among actors in the ecosystem) are identified as units of analysis. The units of analysis are further decomposed into concrete factors or measurements, that are specified later on in this case study protocol. Translated to the research questions, the network structure unit of analysis is related to the first, third and fourth sub research questions, whereas the ecosystem composition is addressed by means of the second sub research question.

A.3 Case Selection

As the main goal of this case study research is uncovering the influence of respectively, the characteristics of the firm ecosystem and the platform type on the network structure of individual platform ecosystems, a first requirement is the variety in the total dataset. The research design encompasses two variables: (1) the characteristics of the firm ecosystem and (2) the type of platform. Accordingly, comparability is achieved by fixing one of the variables whereas the other variable changes for the different case studies. Hereof follows that platforms of two different companies must be selected and that two different types of platforms must be selected for each company to achieve cross-wise comparability.

Additionally, each individual platform that is selected as a case study subject, has to adhere to the following inclusion criteria:

1. The platform of interest is available in multiple regions.
2. The platform and surrounding ecosystem of interest are documented on a dedicated portal, website or platform documentation.
3. Data and meta-data about actors that are part of the platform ecosystem must be retrievable by means of an automated scraping tool.

A.4 Data Collection

To establish triangulation of evidence, a combination of data collection methods will be employed to describe the four case study subjects (e.g. platforms). A combination of a document study, data mining on extension markets and manual inventorization of relationships will be carried out.

A.4.1 Document Study

To explore the platform and its characteristics, a document study will be executed. Documentation found on websites and dedicated portals, will be used to divide platform ecosystems into subsystems. Furthermore, the document study provides for a characterization of a platform in terms of its openness, for example, in relation to platform governance. All relevant documents will be stored in a case study database for later reference. All characteristics for each individual platform will be stored in a spreadsheet, containing references to its respective sources to provide for a transparent chain of evidence and cross-referencing. Among the explored documentation will be documents on the following aspects of each individual platform:

- Platform websites
- Extension markets
- Technical documentation on extension architectures

A.4.2 Data Mining

To identify the complementors of a platform and the relationships among them, a web crawler is employed. The web crawler is derived from an existing tool developed by (Burkard et al., 2011), and then further tailored to suit the needs of this research project. The crawler is Java-based and has a layered architecture. A universal layer provides all data handling functionality, while an extension market specific layer implements the necessities to extract and process all application specific metadata from each extension market. The automated data collection proceeds in two stages, that can be described as follows:

- **Data extraction:** The implemented crawler collects raw data from the extension markets. To start, the web crawler assembles all information about the app categories present in the extension markets. Depending on the architecture of the platform, web addresses of the app information pages are retrieved either from a complete list of apps (Google Apps) or per app category (Office365, Internet Explorer and Google Chrome). Afterwards, all app information pages are traversed to extract the raw data they contain.
- **Data parsing:** The raw data from the app information pages has to be parsed, so that distinct application specific metadata can be saved into a central MySQL database. The raw data is parsed in accordance with predefined template patterns, or Document-Object-Models (DOM). These patterns define how information about an app is presented in the app store. Template patterns, for example, have to be created to derive and process the category, current version, developer, price, description, download count and end-user review information for each application. The pattern templates are different per app store as the data representation differs per app store, and packages such as NekoHTML[1] and Watij[2] are used in the conversion process, depending on if the extension market is AJAX-based or not.

[1]http://www.nekohtml.sourceforge.net
[2]http://www.watij.com

Contrary to the identification of actors, the mapping of relationships between complementors cannot be automated by means of a tool (Iyer, 2012). Therefore, relationships are manually identified by exploring the company website of each individual actor. Company websites contain overviews of alliances, partnerships and other initiatives in which an actor takes part, all of these are registered in a matrix. Additionally, business databases such as CrunchBase[3] are queried with each individual company name to identify additional relationships, with either partners or competitors. Relationships are classified by means of a dummy variable, 0 is used for non existing relationships and 1 for business relationships. Furthermore, relationships are treated as bonded-ties.

A.5 Analysis

The case analysis in this research project will consist of two steps. First of all, results from each individual case study will be reported in a concise case study report, that describes each distinct platform. Each case study report contains a description of the platform, a representation of the ecosystem by means of a layer map and network graph, and a classification of actors that are active in the ecosystem. The following summary provides more detail on each individual component:

- **Platform description:** A description and high level overview of the platform.
- **Adjacency matrix:** A table that contains an overview of all existing relationships between pairs of actors in the ecosystem.
- **Network graph:** A graphical representation of the complementor ecosystem that is constituted of all complementors and the relationships among them.
- **Actor classification:** A classification of important actors in the ecosystem based on their role in the network. The employed actor categorization will be defined earlier in the research project based on findings from a literature study.
- **Network structure:** A characterization of the complementor ecosystem based on a number of network metrics, which are defined previously in this research project based on findings from literature.

Hereafter, cross-wise comparison will be performed to uncover the influence of the firm ecosystem and the type of platform on the structure of an individual platform ecosystem. A comparative analysis will be performed based on observation, for example by means

[3]http://www.crunchbase.com

of visual comparison of stack views, and by means of statistical comparison of network metrics, to identify significant differences.

First, this comparison will be performed to determine the influence of company characteristics on a platform (e.g. a comparison of different platform ecosystems from the same platform owner). Afterwards, similar platforms from different platform owners will be compared to examine the influence of the platform type on individual platform ecosystems. Network analysis is performed with a network visualization tool called Gephi[4] and UCINET[5]. Afterwards, findings from cross-case analysis will be evaluated by means of existing literature, to identify corresponding or contradicting findings.

A.6 Plan Validity

This case study protocol has been formulated in accordance with the case study procedures as defined in the case study checklist by Host and Runeson (2007) and Runeson and Host (2009). All subsequent research steps are based on the theory building case study approach introduced by Eisenhardt (1989), expanded by the principles described by Yin (2009). In addition, validity procedures have been defined to further enhance the validity and reliability of this case study research.

To enhance construct validity, a literature review has led to the identification of network metrics, and their application to software ecosystems. These metrics will be used to compare the network structures of the four platform ecosystems. Draft case study reports will be subject to a comparison with reporting that can be found in similar existing studies, such as the studies conducted by Basole (2009), Iyer (2012), Iyer et al. (2006), Kabbedijk and Jansen (2011) and Madey et al. (2004) to verify the completeness of each case study report. Additionally, as previous studies conducted by Basole (2009) and Iyer et al. (2006) proved that the reconstruction of relationships among actors in the ecosystem is difficult, different sources are used in the data collection process, as relationships are identified from company websites and business databases.

Internal validity is addressed in defining the (sub) research questions that are to be answered in this research project. All (sub) research questions have been based on propositions that have been introduced in previously published scientific literature. Also, as the case analysis considers two variables – being the type of platform and the organizational characteristics of an organization – separate sub research questions have been introduced to examine the effect of one of the variables, while controlling the other one, thus preventing it from exerting a mitigating effect.

[4]http://www.gephi.org
[5]https://www.analytictech.com/ucinet

External validity is enhanced by choosing for a multiple case design, in which each case study has to be selected in accordance with predefined case study selection criteria. Additionally, after the data analysis is performed, a theoretical evaluation is to be performed, as comparison of empirical findings with contradicting or corresponding findings from existing literature increases external validity (Yin, 2009). Reliability then, is ensured by means of this case study protocol, and by storing all case study documents and findings in one central case study database, from which each claim can be traced back to its originating source.

A.7 Study Limitations

Apart from the validity procedures that were elaborated in the previous section, this case study regards limitations that are inherent in the problem domain. A pivotal part of reconstructing the four platform ecosystems is the reconstruction of relationships among the complementors in the ecosystem. While this part of the data collection process is based on similar studies performed by Basole (2009), Burkard et al. (2011), Iyer et al. (2006) and Iyer (2012), it is not possible to provide an encompassing overview of all relationships and dependencies that exist in the ecosystem. As acknowledged by Baldwin and Woodard (2009), network graphs are suitable to provide insight into the business relationships that exist in the ecosystem, while the network graphs fails to provide insight into dependencies among different complements. Therefore, this case study only directs attention at business relationships, and thus leaves other dependencies unaddressed.

Appendix B

Cloud Office Suite Platform Comparison Report

In return for their participation in the validation of the data collection method, Google Apps complementors received a research summary report. The report contained a description of the Google Apps and Microsoft Office365 ecosystems, along with a summary of the comparative analysis. The remainder of this appendix presents the report that was sent out to the Google Apps complementors.

Exploring Platform Ecosystems: A Comparison of Complementor Networks and their Characteristics (Summary)

Joey van Angeren
Department of Information and Computing Sciences
Utrecht University
J.vanAngeren@uu.nl

October 2013

Executive Summary

This report presents a summary of part of a comparative study of proprietary platform ecosystems of Microsoft and Google, conducted by researchers of Utrecht University. It includes a description, visualization and comparison of the ecosystem of third-party application developers and the interfirm relationships among them that surround Google Apps and Office365, two comparable cloud office suite platforms that compete head-to-head in the same market. The aim of this research was to uncover the factors that shape the network structure and interactions among developers in these ecosystems. The report uses data that has been collected at *12-02-2013* by means of automated app sotre data extraction and manual compilation of partner listings for every identified developer with information from their own website.

Results show that both ecosystems are highly centralized and reflect limited interaction among complementors. In addition, the production scope of individual developers appears to be narrow. The Google Apps ecosystem consisted of 992 developers that produced 1341 applications and iniated 1248 interfirm relationships, the 1204 applications for Office365 were developed by 550 third-party developers that initiated 787 interfirm relationships. Google was found more liberal in imposing entry barriers compared to Microsoft, as Microsoft subjects submitted applications to complementary value evaluation in before inclusion in to the app store. A subsequent comparative analysis accompanied by statistical inference brought the following factors forward as an explanation for the recorded differences.

- **Entry barriers:** more liberal entry barriers will increase the size of the ecosystem while the network density decreases.
- **Increasing production activity:** the increasing production actiivity of a developer is positively related to its emergence to initiate interfirm relationships.
- **Developer lock-ins:** the active retention of developers, for example by means of active partner enablement, will result in more densely connected ecosystems while the production scope of individual developers remains unaffected and narrow.

The results presented in this report suggest that the network structure of a platform ecosystem will reflect characteristics of the imposed strategy by the platform owner. Consequently, platform owners can also have influence on the extent at which third-party developers interact with and among each other. However, results also show that the production scope of developers remains narrow, regardless of the imposed regulations around the platform.

Contents

Chapter 1

Introduction

The rise of platforms in high-technology industries has a profound impact on the increasing interconnectivity between companies. In the software industry, thousands of companies assemble around industry platforms such as iOS, Windows 8 or the PlayStation 3 to develop complementary products, that extend the functionality of the platform [7]. While initially a symbol of the mobile era, the enterprise application market recently sees an uptake in the number of app stores that complement existing platforms or suites. The foundation technology of the industry platform exhibits elementary or generic functionality, and provides an extension architecture by means of interfaces (e.g. APIs). Third-parties are responsible for the development of complementarities that make the platform appealing to end-users [1]. A platform owner thus depends on an interlinked ecosystem of developers of software products [10].

To foster innovation, exchange and collaboration around its platforms, platform owners are argued to stimulate the initiation of interfirm relationships among third-parties [8]. Examples of interfirm relationships are alliances, technological partnerships, marketing partnerships, and collaborative research and development. Platform owners may, for example, stimulate intense collaboration through active partner enablement, enforcing platform exclusivity (e.g. lock-ins) or by imposing strict entry barriers for new entrants. Developers of complementarities in turn have their own rationale for the initiation of interfirm relationships. New entrants may choose for technological partnerships with established developers to increase their visibility in the market or to establish chains of interoperability, incumbent firms may extend their number of partners to increase their influence in the ecosystem and to ease their access to resources.

While platform strategists are increasingly involved with the development of platform strategies, little is known about their impact on the network structure of proprietary platform ecosystems. Furthermore, ecosystem visualization studies predominantly address open ecosystems due to their relative transparency. To address this deficiency, Utrecht University visualized, analyzed and contrasted multiple proprietary platform ecosystems of Microsoft and Google. The goal was to increase understanding the complex network structure that characterizes proprietary platform ecosystems, yet at the same time contrasting ecosystems to uncover their principal similarities and differences. For practitioners that develop complementarities, the results provide means to evaluate their own structural position within the ecosystem, and a sense of awareness of how to incorporate a relational strategy in their corporate strategy. This report contains a summary of part of the research findings, and includes a description of the Google Apps and Office365 ecosystem, followed by an enumeration of their principal similarities and differences.

The remainder of this report continues with a description of the method that was used for this research in Chapter 2. Chapter 3 describes and visualizes the ecosystem of developers of complementarities that exists around Google Apps and Office365. Thereafter, Chapter 4 summarizes the main similartieis and differences among these ecosystems, followed by an overview of conclusions an recommendation in Chapter 5.

Chapter 2

Research Approach

The developers of complementarities, the interfirm relationships among them and the characteristics of the Google Apps and Office365 ecosystem were analyzed by means of a case study. Information about the currently available complementarities for Google Apps and Office365 could be found in their app stores, respectively being Google Apps Marketplace[1] and Office365 Marketplace[2]. From the app stores, only software complementarities were considered. For Google Apps software complementarities were listed under the category *"Products"*, and Microsoft categorized software complementarities as *"Applications"*.

The app stores hold application specific metadata for each complementarity, including information about the developer, pricing model, certification status, partner information, and reviews. As this metadata is available on a dedicated website, it could be retrieved by means of a web crawler. The crawler is Java-based and has a two layer architecture. The universal layer is responsible for all data handling. The second layer provides data retrieval from the marketplace. On *12-02-2013*, all application specific metadata from both app stores was retrieved in two stages. First, the crawler traversed the entire list of complementarities in the app stores over multiple iterations to the point at which no new applications were found. Second, all application specific metadata could be read-in through their respective URLs. All data was stored in a central case study database in accordance with predefined pattern templates.

Contrary to the application specific metadata, the interfirm relationships between developers of complementarities could not be retrieved automatically. However, relationships could be retrieved from company websites, partner portals, company news feeds, and openly accessible databases. All complementors were identified by means of an SQL query on the database with vendor data. The company website of each developer was traversed to identify the partnerships they maintain with other developers in the same ecosystem. In addition, CrunchBase[3] was queried to identify additional interfirm relationships. Relationships were regarded as undirected; meaning that if NetSuite lists a relationship with Box, Box is also considered to have a relationship with NetSuite, even if Box omitted to list NetSuite as a partner.

[1]http://www.google.com/enterprise/marketplace
[2]http://office365.pinpoint.microsoft.com
[3]http://www.crunchbase.com

Chapter 3

Results

This chapter presents a description of the Google Apps and Office365 ecosystems. Each ecosystem is visualized, its descriptives are summarized and its main structural properties exploreed. The Google Apps ecosystem is elaborated upon first, followed by a shorter summary of the Office365 ecosystem.

3.1 Google Apps

On *12-02-2013*, the Google Apps Marketplace contained a total of *1354* applications listed under the category *'Products'*, developed by *993* different vendors or individuals. Out of the 1354 applications, *thirteen* are developed by Google, using the aliases *Google Inc.* and *Google Labs.* The *992* remaining developers of complementarities produced *1341* applications, with contributions varying between *one and fifteen* applications. On average, each developer lists *1.36* complementarities with a standard deviation of *0.61.* The vast majority (83%) of the ecosystem consists of developers that list one application. A complete overview of vendor distribution and the number of complementarities they develop is shown in Table 3.1.

Table 3.1: Distribution of Google Apps developers based on the number of applications developed

# of applications	15	14	11	10	9	8	7	6	5	4	3	2	1
# of complementors	1	2	1	1	2	2	4	5	6	10	26	104	826

Google Apps developers limitedly participate in the Google Enterprise Partner [1] partnership model and certification programs employed by Google. To date, *71 (7.16%)* developers obtained certification for reselling services or additional service provision, while *fourteen (1.41%)* vendors are currently listed Google Enterprise Partner. Participation in both programs accounts for a mere *7.36%* of the total ecosystem, as *73* Google Apps developers participate in one or both programs.

According to the data collected from company websites and CrunchBase, the ecosystem is connected by *1248* visual relationships. Actors are connected to one another with an average of *1.26* relationships per actor, which results in a sparsely connected hub-and-spoke network. This hub-and-spoke network is shown in Figure 3.1. Node sizes in the figure are representative for the number of applications developed. Google is regarded as the central player responsible for employing the initiatives in the ecosystem. With a network density that is lower than *1%* (i.e. less than 1% of the relationships that could theoretically be present are initiated), however, the Google Apps ecosystem has a sparse network structure. The majority of developers are new entrants, individuals rather than enterprises, limit themselves to the development of their own applications, or maintain a focused relational portfolio rather than engaging in multiple interfirm relationships.

As shown in Figure 3.1 there is a limited number of lateral connections among complementors in the Google Apps ecosystem, the majority of which is concentrated in the bottom right of the figure. The ecosystem harbors a small group of influential (i.e. networked) developers in the ecosystem,

[1]http://www.google.com/enterprise/gep

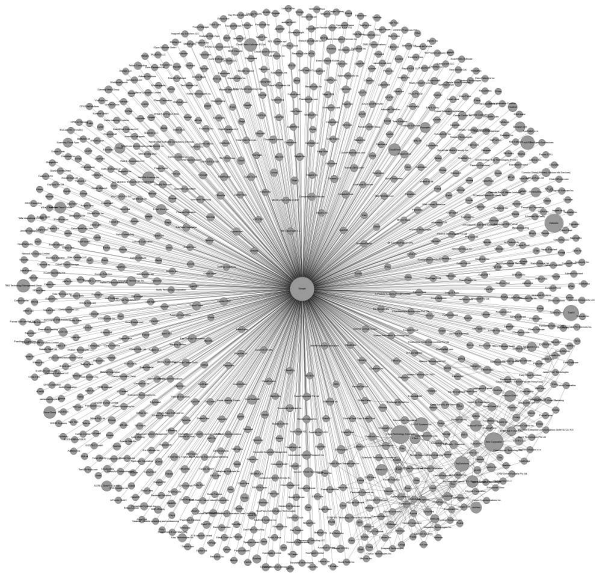

Figure 3.1: Visualization of the Google Apps ecosystem

whereas *73%* of actors are solely connected to Google.*Salesforce.com* is the most embedded actor with *35* relationships, followed by *Zoho Corporation* with a relationship count of *20*, and *Box* that has *20* relationships after which comes *Ping Identity Corporation* with *16* relationships. The dataset is 'cleansed' to provide additional insights. First, to uncover the topology beneath the hub-and-spoke network, Google has been removed from the visualization. Second, complementors solely connected to Google have not been included in the graph. The resulting network has been subjected to the modularity algorithm [2] to divide the network into sub groups, being groups of actors that are more densely connected to one another than to the others in the network. In total, *twelve* clusters were found of which the members are grouped accordingly in Figure 3.2.

Observable is the difference in cluster sizes. There are four clusters that consist of two developers that interact with each other, but not with the other actors in the ecosystem. The small clusters consist of companies that are headquartered at the same geographic location. *Sateraito* and *topgate.co.jp*, for example, are both Japanese whereas *Bittle SAS* and *FITNET APPLICATION* are French. The larger clusters reflect more differences. The *Salesforce.com cluster* has a hub-and-spoke network topology, due to the platform efforts of *Salesforce* itself. Similarly, companies from different geographic locations are grouped around *Zoho Corporation*, making the clusters of *Salesforce.com* and *Zoho Corporation* examples of technology clusters. Other clusters, such as the one in the upper left corner with *OrangeScape* in the middle, lack one central actor and seem to be merely groups of actors connected to one another through direct and indirect relationships. Noteworthy is the absence of a couple of large actors within the ecosystem. *SaaSt* at present offers eleven applications for Google Apps, yet they are not connected to any other complementor in the ecosystem. Similarly, *myERP* is absent while it currently lists nine applications in the Google Apps Marketplace. Other large complementors such as *Zoho Corporation* (15 applications), *Cloud Technology Solutions* (14 applications) and *TopSolutuions* (10 applications) are among the most central actors in the ecosystem.

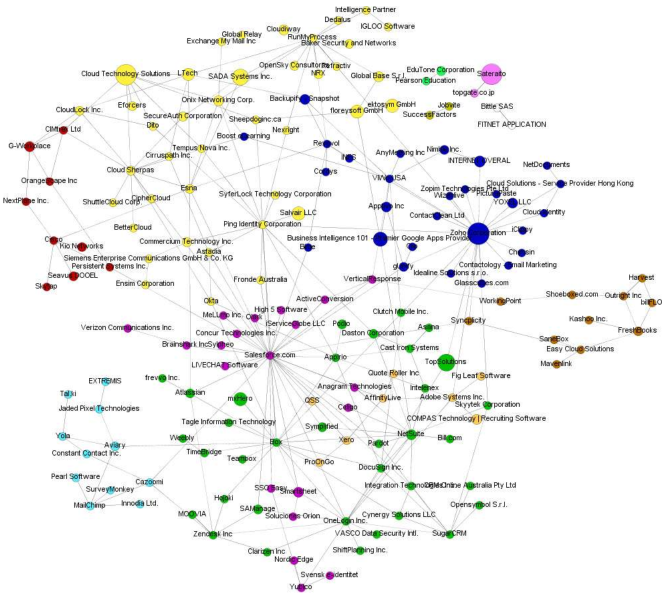

Figure 3.2: Cluster visualization of the Google Apps ecosystem

3.2 Microsoft Office365

Microsoft Office365 is a cloud based office suite platform. For professional use, Office365 bundles scalable versions of Microsoft Office, Microsoft Lync, Microsoft Exchange and SharePoint. Examples of complementarities range from integrations and add-ons to business templates for Microsoft Office products. Complements can be either listed globally *(45,5% of the applications)*, or can be listed in one of the fifty-nine regional marketplaces. At present, Microsoft does not provide an API for Office365. Instead, all components of Office365 offer their own extension mechanism. Microsoft Exchange and Lync have ready-to-use APIs, while Microsoft offers SDKs for complementors that focus on extensions for Sharepoint or Office. At *12-02-2013*, *550* vendors developed one or more applications for Microsoft Office365. Of the 550 developers of complementarities *278 (50,50%)* are Microsoft partners. Noteworthy is that Microsoft itself does not develop applications for Office365. Accordingly, all *1204* applications are developed by third-parties. On average, each developer of complementarities develops *2.18* applications with a standard deviation of *1.65*. The largest vendor in the ecosystem lists *39* applications, while *67%* of vendors only develop one application. A complete distribution of these figures is included in Table 3.2. Important to note is that the actual number of unique complements for Office365 may be lower than the number found in this research because Microsoft lists complementarities for Microsoft Office per component of the suite..

Table 3.2: Distribution of Office365 developers based on the number of applications developed

# of applications	39	32	23	22	16	14	13	11	10	9	8	7	6	5	4	3	2	1
# of complementors	1	2	1	1	1	1	1	3	3	5	4	7	13	10	23	24	82	368

The Office365 developers are connected by *787* relationships, accounting for an average of *1.43* relationships per developer with a standard deviation of *11.74* relationships. The most well connected actors are *Dell* with *37* relationships and *Nintex* with *32* relationships that are among the largest vendors in the ecosystem, followed by the small vendors *AvePoint* and *K2* with *23* and *16* relationships. The network density equals a value of *0.5%*. Consequently, the centralization score for the Office365 ecosystem is *99.84%*, lower than the previously discussed ecosystem. The network graph of the entire network is shown in Figure 3.3.

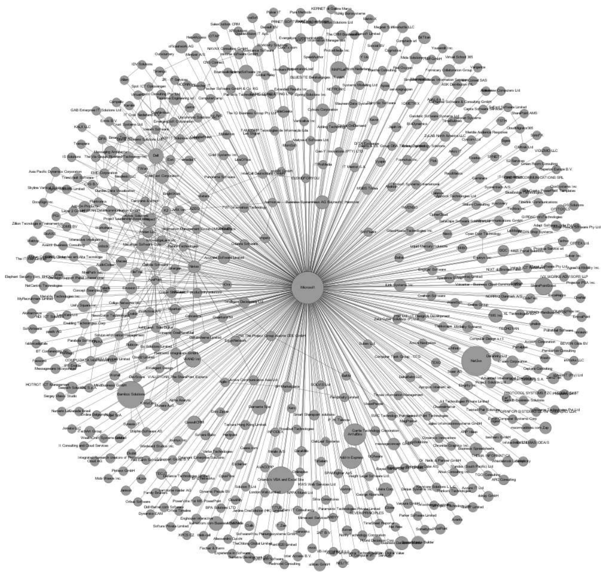

Figure 3.3: Visualization of the Office365 ecosystem

In total, 164 out of 550 developers *(29,82%)* engage in partnerships with other Office365 developers. This interconnectedness may be partially explained by the presence of established Sharepoint developers in the ecosystem, as the on premises version of the product already came with an extension architecture. The 164 complementors are distributed over *eighteen* clusters, shown in Figure 3.4. The network topology in Figure 3.4 is constituted of either large or very small clusters. The largest cluster in the ecosystem is visualized in the upper left corner of the graph, showing the developers of complementarities that have intensive technological ties with *Dell*. However, these developers do not seem to intensively interact with each other, opposed to the actors that surround *Nintex* and *AvePoint*. The *Nintex* cluster seems to embody the most interconnected part of the ecosystem, populated by relatively large vendors.

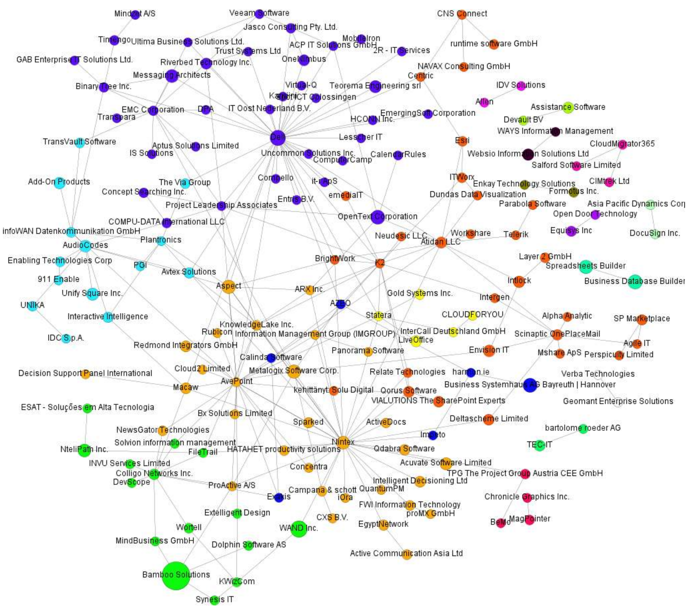

Figure 3.4: Cluster visualization of the Office365 ecosystem

Chapter 4

Analysis

This chapter presents the results a cross-case analysis of the case studies that were described in the previous chapter. While the previous chapter elaborated upon each case in isolation, commonalities stand out. First of all, both platform ecosystems exhibit the same structural properties in their global network topology. The platform ecosystems are sparsely connected and highly centralized hub-and-spoke networks. Interfirm relationships are limitedly initiated among actors in the ecosystem, indicating a focused relational strategy of firms under increasing specialization or a lack of interest in initiating collaboration. Second, the majority of developers in all ecosystems *(between 67% and 83%)* develop one application. The large number of small developers could be explained by the relative novelty of online marketplaces with complimentarities or the ongoing search for viable business models to exploit the new app industry paradigm. These circumstances give rise to increased opportunism in the ecosystem, certainly as start-up investments are next to zero [4]. Last, each platform seems to have imposed at least an extent of entry barriers to inclusion of new complementarities in the app store.

Despite the apparent similarities, the platform ecosystems can also be contrasted to identify noteworthy differences. Table 4.1 summarizes the principal descriptives for each of the four platform ecosystems. The descriptives summarized in Table 4.1 will be used for the cross-case analysis presented in this chapter. Cross-case analysis aims for the synthesis of one or more theories in an iterative process. The presented propositions are interpreted as a formalization of observed effects, and as such need statistical interference in future work to enhance their reliability and generalizability. The remainder of this chapter contrasts the four platform ecosystems from different perspectives. It addresses the perspectives of entry barriers, likelihood of interfirm relationship initiation and partnership model participation.

Table 4.1: Descriptives of the four platform ecosystems

Characteristic	**Google Apps**	**Office365**
Owner	Google	Microsoft
Entry barriers	Technical validation	Compatibility and complementarity value validation
Ecosystem size	993	551
Partner coverage	1.41%	50.50%
Avg. applications per developer	1.36	2.18
Standard deviation for avg. applications	0.61	1.65
Developers with one application	83.18%	66.85%
Avg. relationships per developer	1.26	1.43
Avg. relationships per developer (excl. platform owner)	0.258	0.431
Standard deviation for avg. relationships	15.74	11.74
Standard deviation for avg. relationships (excl. platform owner)	1.01	1.37
Developers with relationships	17.14%	29.82%
Network density	0.25%	0.50%
Centralization	99.94%	99.84%

4.1 Entry Barriers

Openness and entry barriers are perhaps one of the most recurrently addressed aspects of platform strategy. Google Apps and Microsoft Office365 can be contrasted based on their degree of architectural control, the extension architectures in place and entry barriers to entering app stores. With regard to architectural control, Google Apps and Office365 are considered comparably 'closed'. With regard to entry barriers to the app store, the barriers imposed by Google appear more liberal than those of Microsoft. Google Apps and Office365 are comparable in their architectural requirements, however, Microsoft imposes discriminatory complementary value requirements. With these requirements in hand, Microsoft reserves the right to exclude any application from the app store that does not provide direct added-value to the platform.

Apparent from the ecosystem descriptives presented in Table 4.1 is the higher platform adoption by application developers of Google Apps opposed to its Microsoft equivalent. The Google Apps ecosystem is larger than its Microsoft counterpart, with a difference of *440* developers. This higher rate of adoption coincides with more liberal entry barriers. More concrete, Google sees two to three times higher adoption rates for their platforms with more liberal governance. Accordingly, empirical evidence postulates that lowering entry barriers to a platform ecosystem will be positively related to the adoption of the platform.

Proposition 1: Lowering entry barriers to a platform ecosystem will be positively related to the number of application developers.

This proposition finds supportive evidence in existing scientific literature. West [15], Jansen, Brinkkemper and Finkelstein [11] and Cusumano [5] draw parallels between increased openness and increasing rates of adoption or ecosystem growth. With their 'open enterprise model', Jansen, Brinkkemper, Souer and Luinenburg [12] envisage a need for increased openness by providing software producing organizations with a model to determine how 'open' or 'closed' organization is and guidelines to increase openness. The authors argue that product or platform selection are based on prejudice with regard to openness and transparency. As Jansen et al.[12] reason, a lack of transparency and openness may harm the reputation of a platform owner and its ecosystem, for they are deemed to foster lock-ins and abue of intellectual property rights. Boudreau [3] provides empirical support for the positive relationship between lowered entry barriers and increasing adoption from the handheld computer industry. The author records that an increase in the number of licensed complementary hardware suppliers coincides with growing adoption of the technology platform. Despite the supportive evidence, caution is needed in assuming an exclusive causal relationship between entry barriers and platform adoption. Part of the increased adoption of the Google platforms may also be attributed to bigger popularity and thus the advantage of critical mass, superior proprietary technology in a young market or Google more actively subsidizing the supply side of the market [13, 6].

4.2 Development Activity and Embeddedness

Despite the greater number of developers around Google Apps, the average number of applications for the Microsoft platforms is higher. Every Google Apps developer on average develops *1.36* complements, in contrast the average number of application per developer amounts *2.18* for Office365. Apart from higher development activity, the Office365 ecosystem also appears more interwoven and thus more robust compared to their Google peers. At present, *29.82%* of Office365 developers engage in at least one interfirm relationship with another developer, compared to *17.14%* of the Google Apps developers. Consequently, the average number of interfirm relationships per developer is also higher in Office365.

The coincidence of increasing development activity and higher interconnectivity in the ecosystem gives rise to the hypothesis that increasing commitment of a developer to the ecosystem and its emergence to collaborate with other developers in the ecosystem are positively related. This hypothesis builds on the notion that power and influence are inherently relational in social and business networks alike [9, 10]. It assumes that principal developers in the ecosystem will search for interfirm relationship to ease their access to resources and means to exert power in the ecosystem. Simultaneously, smaller actors will search for technological integrations with complementary products to increase their visibility in the market or extend their potential user base.

Correlations

		Complements	Relationships
Complements	Pearson Correlation	1	,232**
	Sig. (2-tailed)		,000
	N	992	992
Relationships	Pearson Correlation	,232**	1
	Sig. (2-tailed)	,000	
	N	992	992

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 4.1: SPSS output for Person correlation between number of complements developed and number of interfirm relationships in the Google Apps ecosystem

To examine the existence of the hypothetical relationship between development activity and embeddedness in a platform ecosystem, a Pearson correlation analysis is performed. It investigates the existence of a casual relationship between the development activity of a developer and the number of interfirm relationships in its relational portfolio. In the correlation analysis, the number of applications per developer is considered as the independent variable and the number of interfirm relationships initiated as the dependent variable. The result of the correlation analyses is shown in the SPSS output fragments, which can be found in Figure 4.1 for Google Apps and in Figure 4.2 for Office365.

Correlations

		Complements	Relationships
Complements	Pearson Correlation	1	,131**
	Sig. (2-tailed)		,002
	N	550	550
Relationships	Pearson Correlation	,131**	1
	Sig. (2-tailed)	,002	
	N	550	550

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 4.2: SPSS output for Person correlation between number of complements developed and number of interfirm relationships in the Office365 ecosystem

In both ecosystems, there is a significant (*0.01* significance level) correlation between the number of applications developed and the number of interfirm relationships initiated. For Google Apps, the obtained value equals *0.232* while for Office365 a slightly lower value is found (*0.131*). Statistical evidence thus suggests a causal relationship between increasing development activity and the embeddenss of a developer in a proprietary platform ecosystem. The positive correlation values obtained suggest that the direction of this relationship is positive. Empiricial evidence thus suggests that increasing development activity coincides with increasing emergence to initiate interfirm relationships in a proprietary platform ecosystem. Consequently, the following proposition is formulated:

Proposition 2: An increase in the number of applications a developer creates will be positively related to the number of interfirm relationships it initiates in a platform ecosystem.

An explanation for the mild correlation can be sought in the number of new entrants that already initiated interfirm relationships with other actors in the Google Apps or Office365 ecosystem, in before actually joining the ecosystem. For example, a Zoho Corporation partner that enters the Google Apps ecosystem is better embedded compared to a new start-up. Other companies may first focus on enriching their product portfolio rather than an extension of their relational portfolio. An additional explanation for the mild correlation obtained from the Office365 ecosystem is the presence of a substantial amount of SharePoint developers. These developers are likely to have established interfirm relationships long before the introduction of Office365.

4.3 Partnership Model Participation

One can argue that the active partner enablement strategy of Microsoft has a profound impact on the productivity and robustness of Microsoft platform ecosystems. Microsoft can foster lock-in effects, enforce platform exclusivity for applications [7] and further increase (quality) control through certification by means of their partnership model [14]. The Microsoft Certified Partner Network functions as an umbrella program for partnership models and certification schemes, and has a high coverage across the wider Microsoft ecosystem. In the Office365 ecosystem for instance *50.50%* of actors are certified as Microsoft partners. In contrast, the partner coverage in the Google Apps ecosystem equals *1.41%*.

Group Statistics

	PartnerStatus	N	Mean	Std. Deviation	Std. Error Mean
Complements	1,00	278	2,3129	3,35061	,20096
	,00	272	2,0625	3,38977	,20554
Relationships	1,00	278	1,1942	2,92475	,17542
	,00	272	,5221	2,50893	,15213

Figure 4.3: Group statistics of Microsoft partners and non Microsoft partners in the Office365 ecosystem

To examine whether partners are more active in developing applications and initiating interfirm relationships compared to non partners, an independent sample T-test is performed. Because limited partner coverage hampers the equal distribution of developers in two groups, only the Office365 ecosystem can be subjected to statistical interference. Office365 developers are distributed in two groups based on their current partner status. Partner status is coded as a dummy variable, in which '1' corrsponds to a status as Microsoft partner and '0' represents non partners. Out of the *550* developers, *278* are acknowledged as Microsoft partners and *272* are categorized as non partners. The independent sample T-test was performed at the *95% confidence interval*. Figure 4.3 shows the SPSS output of the computed group means. A Microsoft partner in the Office365 ecosystem on average initiated *1.192* interfirm relationships with a standard deviation of *2.925*, a non partner on average had *0.5221* ties with a standard deviation equal to *2.509*. The partner category has a more skewed distribution of interfirm relationships compared to the non partners category.

Based on these figures, there is a significant difference in embeddedness between partners and non partners; $t(538) = 2.895$, $p = 0.004$ (the full SPSS output for the independent samples T-test is included in Figure 4.4). These results suggest that Microsoft partners in the Office365 ecosystem have significantly more interfirm relationships compared to non partners. Empirical evidence postulates the proposition that active partner management can foster the interconnectivity in the ecosystem.

Proposition 3: More active partner enablement will be positively related to the network density of a platform ecosystem.

The second independent sample T-test is performed to investigate the relationship between partnership model participation and development activity. Microsoft partners on average develop *2.313* applications with a standard deviation of *3.351*, and non partners develop *2.063* applications with a standard deviation equal to *3.390*. Based on these group means, there is no significant difference in development activity between partners and non partners; $t(548) = 0.871$, $p = 0.384$. Results imply that Microsoft partners are not significantly more committed to development of complementarities compared to non partners, and that the small difference in group means may be attributed to random variation or luck.

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means	
		F	Sig.	t	df
Complements	Equal variances assumed	2,000	,158	,871	548
	Equal variances not assumed			,871	547,387
Relationships	Equal variances assumed	8,403	,004	2,890	548
	Equal variances not assumed			2,895	538,817

Independent Samples Test

		t-test for Equality of Means		
		Sig. (2-tailed)	Mean Difference	Std. Error Difference
Complements	Equal variances assumed	,384	,25045	,28741
	Equal variances not assumed	,384	,25045	,28745
Relationships	Equal variances assumed	,004	,67219	,23258
	Equal variances not assumed	,004	,67219	,23219

Independent Samples Test

		t-test for Equality of Means 95% Confidence Interval of the Difference	
		Lower	Upper
Complements	Equal variances assumed	-,31412	,81502
	Equal variances not assumed	-,31419	,81509
Relationships	Equal variances assumed	,21533	1,12904
	Equal variances not assumed	,21607	1,12830

Figure 4.4: SPSS output of independent samples T test for Microsoft partners and non Microsoft partners in the Office365 ecosystem

Chapter 5

Conclusion and Recommendations

This summary presented the first results of an exploratory research that investigates the factors that shape proprietary platform ecosystems. By means of automated data extraction from app stores and manual identification of interfirm relationships by archiving partnership mentions on the website of developers, the Google Apps and Office365 ecosystems were described and compared. The Google Apps ecosystem harbored 992 developers, that developed 1341 applications and initiated 1248 interfirm relationships. For Office365, these figures did equal 550 developers, 1204 applications and 787 interfirm relationships.

Results showed that both ecosystems are sparesely connected hub-and-spoke networks, in which the majority of developers only created one application. Entry barriers to both app sotres were found to differ, as Microsoft apart from architectural compatibility also assesses the complementary value of new applications, in before they are included in the app store. Meanwhile, the Google Apps ecosystems harbored more application developers, whereas its Microsoft counterpart appeared more cohesive and exhibited higher productivity per complementor. A comparative analysis and statistical interference revealed that incresing development activity and growth in the number of initiated interfirm relationships by a developer coincide, and that partners are more embedded in the ecosystem compared to non partners while their development scope remains unaffected. In sum, the following factors may have an influence on the network structure of proprietary platform ecosystems.

- **Entry barriers:** more liberal entry barriers will increase the size of the ecosystem while the network density decreases.
- **Increasing production activity:** the increasing production actiivity of a developer is positively related to its emergence to initiate interfirm relationships.
- **Developer lock-ins:** the active retention of developers, for example by means of active partner enablement, will result in more densely connected ecosystems while the production scope of individual developers remains unaffected and narrow.

The results presented provide a first step towards better understanding of the dynamics around multi-sided platforms that are increasingly present in the software industry. This ongoing research project aims to compare multiple platform ecosystems to uncover their characteristic similarities and differences.

Bibliography

[1] C.Y. Baldwin and C.J. Woodard. The architecture of platforms: A unified view. In A. Gawer, editor, *Platforms, Markets and Innovation*, pages 19–45, Cheltenham, UK, 2009. Edward Elgar Publishing.

[2] V.D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics*, 5(10), 2008.

[3] K. Boudreau. Open Platform Strategies and Innovation: Granting Access vs. Devolving Control. *Management Science*, 56(10):1849–1872, 2010.

[4] M. Butler. Android: Changing the Mobile Landscape. *IEEE Pervasive Computing*, 10(1):4–7, 2011.

[5] M.A. Cusumano. Technology strategy and management: The evolution of platform thinking. *Communications of the ACM*, 53(1):32–34, 2010.

[6] T. Eisenmann, G.G. Parker, and M.W. Van Alstyne. Opening platforms: How, when and why? In A. Gawer, editor, *Platforms, Markets and Innovation*, pages 131–162, Cheltenham, UK, 2009. Edward Elgar Publishing.

[7] A. Gawer. Platform dynamics and strategies: From products to services. In A. Gawer, editor, *Platforms, Markets and Innovation*, pages 45–76, Cheltenham, UK, 2009. Edward Elgar Publishing.

[8] A. Gawer and M.A. Cusumano. *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation*. Harvard Business School Press, Boston, MA, USA, 2002.

[9] R.A. Hanneman and M. Riddle. *Introduction to social network methods*. University of California, Riverside, CA, 2005.

[10] M. Iansiti and R. Levien. Strategy as Ecology. *Harvard Business Review*, 82(3):68–78, 2004.

[11] S. Jansen, S. Brinkkemper, and A. Finkelstein. Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems. *Proceedings of the First International Workshop on Software Ecosystems*, pages 34–48, 2009.

[12] S. Jansen, S. Brinkkemper, J. Souer, and L. Luinenburg. Shades of gray: Opening up a software producing organization with the open software enterprise model. *The Journal of Systems and Software*, 85(7):1495–1510, 2012.

[13] M.L. Katz and C. Shapiro. Product introduction with network externalities. *Journal of Industrial Economics*, 40(2):55–83, 1992.

[14] J. Van Angeren, J. Kabbedijk, K.M. Popp, and S. Jansen. Managing software ecosystems through partnering. In M.A. Jansen, S. Cusumano and S. Brinkkemper, editors, *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*, pages 85–102, Cheltenham, UK, 2013. Edward Elgar Publishing.

[15] J. West. How Open is Open Enough? Melding Proprietary and Open Source Platform Strategies. *Research Policy*, 32(7):1259–1285, 2003.

Appendix C

Research Papers

This appendix includes the two research papers that were written and published in the process of this thesis. The first paper elaborates upon the analysis of the Google Apps ecosystem, and statistically investigates the relationship between the activity of a complementor (i.e. the number of apps developed) and its emergence to engage in interfirm relationships. The second paper explores the Office365 ecosystem and investigates how active complementor retention influences the network structure of the ecosystem.

Suggested citation

Angeren, J. van, Blijleven, V., Jansen, S., & Brinkkemper, S. (2013). Complementor Embeddedness in Platform Ecosystems: The Case of Google Apps. *Proceedings of the Seventh International Conference on Digital EcoSystems and Technologies*, 37-42.

Angeren, J. van, Jansen, S., & Brinkkemper, S. (2014). Exploring the Relationship between Platform Strategy and Interfirm Network Structure: An Analysis of the Office365 Ecosystem. *Manuscript under review at the International Conference on Software Business.*

Complementor Embeddedness in Platform Ecosystems: The Case of Google Apps

Joey van Angeren, Vincent Blijleven, Slinger Jansen, Sjaak Brinkkemper
Department of Information and Computing Sciences, Utrecht University
Princetonplein 5, 3508 TB Utrecht, the Netherlands
Email: {j.vanangeren,v.b.blijleven,slinger.jansen,s.brinkkemper}@uu.nl

***Abstract*—Platforms and their marketplaces with complementarities are prominent in the software industry. As the proprietary platform itself exhibits elementary or generic functionality, platform owners depend on a complementor ecosystem populated by third-parties. At present, little is known about mechanisms at play in proprietary ecosystems. Addressing this deficiency, this paper investigates the Google Apps ecosystem through statistical and network analysis. Results show that the Google Apps ecosystem is sparsely connected, the majority of complementors develops one application (83%) and does not have visible relationships (73%). Furthermore, there is a positive relationship between the number of applications a complementor develops and the number of relationships it establishes. The research method and results presented can be used by practitioners as a reference to evaluate their structural position in the ecosystem, while it provides researchers with a quantification of ecosystem characteristics.**

I. Introduction

The rise of platforms in high-technology industries has a profound impact on the increasing interconnectivity between companies. In the software industry, thousands of companies assemble around industry platforms such as iOS, Windows 8 or the PlayStation 3 to develop complementary products, that extend the functionality of the platform [1], [2]. An industry platform herein is defined by Gawer [1] 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". The foundation technology exhibits elementary or generic functionality, and provides an extension architecture by means of interfaces (e.g. APIs). Third-parties – from now on to be referred to as complementors – are responsible for the development of complementarities that make the platform appealing to end-users [3]. A platform owner thus depends on an interlinked ecosystem of developers of software products [4], [5], [6].

Platform and ecosystem strategists agree [7], [8], [5] that complementors should be encouraged to establish formal relationships. Relationships such as technological partnerships and alliances drive innovation and collaboration, as increasing interconnectivity fosters the exchange in the ecosystem [7], [8]. Accordingly, platform owners should seek participation of new entrants, stimulating them to develop additional complements. In sum, platform strategies aim to increase the involvement and embeddedness of individual actors in the ecosystem. Embeddedness herein, is the ratio of the number of relationships an actor has with others to the total number of relationships that is theoretically possible [9]. Actors that have many relationships are considered prominent and influential in the ecosystem compared to peripheral peers that only have a relationship with the platform owner to showcase their complements.

While platform strategists are increasingly involved with the development of platform strategies, little is known about their impact on the network structure of proprietary platform ecosystems as longitudinal studies are scarce. Furthermore, ecosystem visualization studies predominantly address open ecosystems due to their relative transparency. To address this deficiency, this paper summarizes the characteristics of the Google Apps ecosystem (a cloud-based office suite) and examines the influence of vendor development activity on their embeddedness in the ecosystem. In order to do so, this paper answers the research question: *"What is the influence of the number of complements developed by an actor on its embeddedness in a proprietary platform ecosystem?"*

This paper presents a visualization and analysis of the Google Apps platform ecosystem by means of social network analysis [9], [10]. The Google Apps ecosystem is visualized based on data collected from its app store (Google Apps Marketplace), and by means of manual reconstruction of formal relationships. Through the analysis of the Google Apps ecosystem, the factors that shape the network structure of platform ecosystems are identified. Furthermore, this paper provides empirical evidence for the recurring hypothesis [7], [8] that there is a relationship between the increasing involvement of an actor in the ecosystem and its emergence to collaborate with other complementors in the same ecosystem. In addition, this paper aims to improve the understanding of mechanisms at play in platform ecosystems, being among the first to visualize the ecosystem that forms around a proprietary platform.

The remainder of this paper continues with an overview of related work in Section II, followed by a description of the research approach in Section III. Section IV presents an overview of the Google Apps ecosystem, containing a network graph of part of the ecosystem. Section V describes the analysis of the Google Apps ecosystem and provides an answer to the research question. In Section VI validity threats are addressed, followed by a conclusion and suggestions for future research in Section VII.

II. Related Work

Software companies engage in alliances and technological partnerships in search for increased interoperability, integration, specialization and influence [11], [8]. Platform owners stimulate this interconnectivity to benefit from network effects such as co-creation and co-innovation by complementors [7],

[1]. Platform owners face complex and far reaching decisions in the management of a platform and its surrounding ecosystem. According to West [12], platform owners face the challenge of balancing appropriability and adoption. This refers to the balance between reaping benefits from a proprietary platform while these benefits are connected to the adoption of the platform by complementors and end users. At the same time, platform owners entice to commit complementors to their platform, for example, through the stimulation of collaboration in the ecosystem [7]. Not only does collaboration foster the diffusion of innovation and productivity in the ecosystem [5], [13], it also provides cohesion in the ecosystem which is pivotal in decision-making about interactions between complements and platform [7], [14].

This paper complements previous work that produced techniques for data extraction from app stores [15], [16], [2], and builds on the body of literature on the visualization of interfirm relationships. Quaadgras [17] and Basole [18], for example, presented results of investigations into the RFID and mobile ecosystems. Basole [18] used centrality and network density metrics [9] to calibrate the ecosystem. The author found that emerging segments in the mobile industry were sparsely connected compared to their more mature counterparts.

Iyer, Lee and Venkatraman [11], visualized the proprietary ecosystems of IBM, Microsoft and SAP between 1990 and 2002. The ecosystems can be visualized as a hub-and-spoke network in which the platform owner is the hub in the center. The platform owner is surrounded by complementors that engage in formal and competitive relationships with the platform owner and each other, providing a coopetitive environment [19]. Between 1990 and 2002, the ecosystems witnessed a stable growth, yet became more sparsely connected as companies were found selective in their partnership activities.

III. Research Approach

The complementors, the formal relationships among them and the characteristics of the Google Apps ecosystem are analyzed by means of a single case study [20]. Google Apps is a cloud based office suite platform, intended for use by enterprises and governmental or educational institutions. The platform consists of scalable versions of Google products, such as Gmail, Google Drive and Google Sites. Apart from its standard functionality, Google Apps depends on complementors that diversify or extend its functionality. The complements – either developed by Google or third-parties – can be found in the Google Apps Marketplace[1]. Only the complements listed under the category *"products"* were considered, meaning that service providers have been excluded.

The Google Apps Marketplace holds application specific metadata for each complement, which among others includes information about the developer, pricing model, certification status, partner information, and reviews. As this metadata is available on a dedicated website, it could be retrieved by means of a web crawler. The crawler used leans on the architecture of a previous initiative by Burkard, Draisbach, Widjaja and Buxmann [15], who in 2010 retrieved application specific metadata from five app stores on a weekly basis. The crawler is Java-based and has a two layer architecture. The universal layer is responsible for all data handling. The second layer provides data retrieval from the marketplace. On *12-02-2013*, all application specific metadata for the Google Apps Marketplace was retrieved in two stages. First, the crawler traversed the entire list of complements in the Google Apps Marketplace over multiple iterations to the point at which no new applications were found. Second, all application specific metadata could be read-in through their respective URLs. The web pages were converted into Document-Object-Models with an open-source tool called NekoHTML[2], as this simplified the parsing of data. Afterwards, all data was stored in a central case study MySQL database in accordance with predefined pattern templates [20].

Contrary to the application specific metadata, the formal relationships between complementors could not be retrieved automatically. However, business and competitive relationships could be retrieved from company websites, partner portals, company news feeds, and proprietary or openly accessible databases [18], [11]. The relationships that can be established within the boundaries of a platform ecosystem are shown in the meta-model in Fig. 1. All complementors were identified by means of an SQL query on the database with vendor data, followed by manual verification because some companies use multiple aliases. Both aliases *'Google Inc.'* and *'Google Labs'*, for example, refer to Google. The company website of each vendor was then visited to identify the partnerships they maintain with other complementors. In addition, CrunchBase[3] was queried to identify competitors and additional partnering relationships. Herein, relationships are regarded as undirected, meaning that if NetSuite lists a relationship with Box, Box is also considered to have a relationship with NetSuite, even if Box omitted to list NetSuite as a partner.

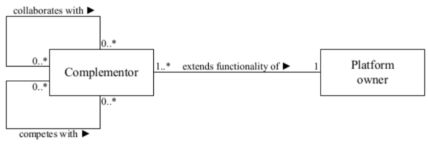

Fig. 1. Meta-model of relationships in a proprietary platform ecosystem

The identified relationships were maintained in an adjacency matrix, created in UCINET[4]. An adjacency matrix is a matrix in which for every pair of actors (e.g. an adjacency pair), the presence or absence of a relationship is denoted [9]. In the adjacency matrix, **-1** was used to indicate the presence of a competitive relationship, **1** was used to indicate the existence of a partnership, and **0** indicated that two actors do not have a relationship. Because of its compatibility with UCINET, Gephi[5] [21] was used to visualize the ecosystem by means of network graphs. The structural properties of the network were computed with either UCINET or Gephi.

To validate the accuracy of the data collection method and to gain insight into the partnering strategies of Google Apps complementors, 35 vendors were contacted by email.

[1] http://www.google.com/enterprise/marketplace

[2] http://www.nekohtml.sourceforge.net

[3] http://www.crunchbase.com

[4] http://www.analytictech.com/ucinet

[5] http://www.gephi.org

The *eighteen largest vendors* and the *seventeen vendors with the most business relationships* were contacted. The complementors were asked whether their relationship portfolio as identified was complete and whether or not they take an active stance towards partnering with other complementors in the Google Apps ecosystem.

IV. Descriptives of the Google Apps Ecosystem

This section provides a description of the Google Apps ecosystem. The units of analysis of the Google Apps ecosystem are the complementors and the relationships among them. Accordingly, the remainder of this section first elaborates on the distribution of complementors, followed by a description of the network structure of the ecosystem.

A. Complementors

On *12-02-2013*, the Google Apps Marketplace contained *1354* applications, developed by *993* different developers. *Thirteen* applications are developed by Google, using the aliases *Google Inc.* and *Google Labs*. The *992* complementors develop *1341* applications, with contributions varying between *one and fifteen* applications. On average, each actor develops *1.36* complements with a standard deviation of *0.61*. The vast majority (83%) of the ecosystem thus consists of complementors that develop one application. A complete overview of vendor distribution based on the number of applications they develop is shown in Table I. Among the complements are integrations with other platforms, data synchronizations, and extensions that facilitate the execution of business processes.

TABLE I. Distribution of complementors based on the number of applications developed

# of applications	# of complementors
15	1
14	2
11	1
10	1
9	2
8	2
7	4
6	5
5	6
4	10
3	26
2	104
1	826

The ecosystem is connected by *1391* formal relationships such as technological partnerships and alliances. Out of the total number of relationships, *143* competitive relationships were identified, accounting for *10.28%* of all relationships in the ecosystem. This finding supports the notion that ecosystems can be regarded as coopetitive [19] environments, where collaboration and competition occurs simultaneously. Furthermore, the overall network is connected with an average of *1.26* relationships per actor, as every complementor is considered to have a business relationship with Google. To stimulate the engagement of actors in the ecosystem, Google employs a partnership model (i.e. Google Enterprise Partners) that crosses the boundaries of the Google Apps ecosystem. In addition, Google offers certification programs for applications or reseller services. Whether a vendor participates in one of these programs is indicated on the Google Apps Marketplace. Participation in both programs accounts for a mere *7,36% (73 vendors)* of the ecosystem.

B. Ecosystem

The Google Apps ecosystem is constituted of actors (nodes) and the relationships among them (edges). Network metrics [22], [9], [10] were used to calibrate the main structural properties of the network. The values found for these metrics are summarized in Table II (network level metrics) and Table III (node level metrics). In sum, the following metrics are used to calibrate the network structure on the network level:

- **Size:** The number of nodes the ecosystem consists of.
- **Density:** The ratio of relationships that are present to the maximum number of relationships that can theoretically be formed in the ecosystem.
- **Centralization:** The ratio of the structural position of the platform owner to the embeddedness values of all complementors.

TABLE II. Network level metrics for the Google Apps ecosystem

Metric	Value
Size	993
Density	0.00282
Density (noncompetitive)	0.00253
Centralization	0.9994

Following a centralization score of *99,94%* – a measure that compares the network to an ideal star-shaped network in which no complementor relationships exist – Google is found to be the central player responsible for employing the initiatives in the ecosystem. With a network density lower than *1%*, the Google Apps ecosystem has a sparse network structure. The majority of complementors are thus new entrants, limit themselves to the development of their applications, or maintain a selective set of key relationships. Since our study shows an increase of *68%* in the number of complementors compared to the figures found by Burkard et al. [15] in November 2010, the presence of a substantial amount of new entrants is apparent. In addition, part of the small complementors (i.e. that only develop one application) are individuals rather than enterprises.

To provide additional insights related to the role of individual actors in the ecosystem, the following network metrics have been used for node level measurements of which the values are presented in Table III:

- **Embeddedness:** The ratio of the number of relationships an actor has to the number of relationships that is theoretically possible.
- **Eigenvector centrality:** The structural position of an actor as a function of its relationship to other actors, weighted by their centralities.
- **Clustering coefficient:** The ratio of the number of partners of an actor that is also connected to one another to the total number of partners an actor has.

TABLE III. Node level metrics for the Google Apps ecosystem

Metric	Min.	Max.	Avg.	Std. dev.
Embeddedness	0.00101	1	0.00282	0.00285
Embeddedness (noncompetitive)	0.00100	1	0.00253	0.00261
Eigenvector centrality	0.0298	1	0.0318	0.00315
Clustering coefficient	0	1	0.193	0.287

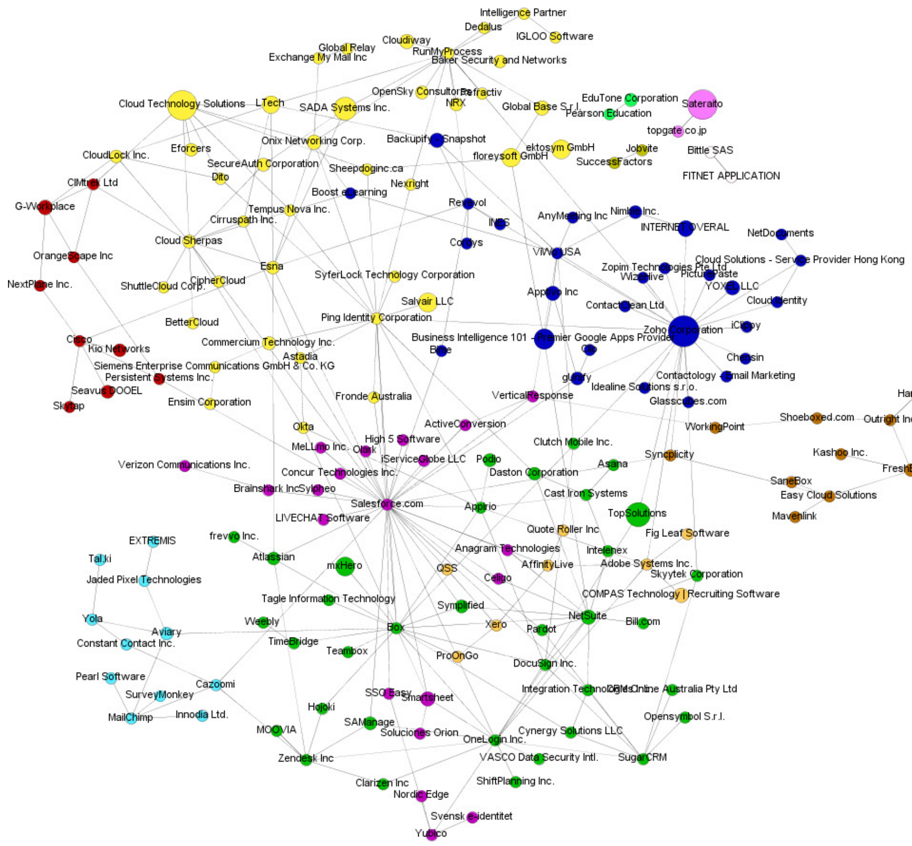

Fig. 2. Network graph showing clusters in the Google Apps ecosystem

The high standard deviation for embeddedness reveals the presence of a small group of influential actors in the ecosystem, while 73% of actors are solely connected to Google. Closer inspection of the dataset confirms this finding, identifying *Salesforce.com* as the most embedded actor with *48* relationships, followed by *Zoho Corporation* with *33* partners, and *Box* that has *21* relationships. Discarding competitive ties, *Salesforce.com* and *Zoho Corporation* lose the most connections, evidencing the presence of competitors. The clustering coefficient reveals the presence of a number of clusters [10] in the ecosystem, being groups of actors that maintain many relationships with each other.

A useful network visualization could be created after 'cleansing' the dataset. First, the remainder of this research excludes competitive relationships. Second, to uncover the topology beneath the hub-and-spoke network, Google has been removed from the visualization. Finally, complementors solely connected to Google have not been included in the graph. The resulting network visualization of the Google Apps ecosystem is shown in Fig. 2. The sizes of the nodes in the graph represent the number of applications they develop for Google Apps.

The ecosystem has been divided in clusters by means of the modularity algorithm in Gephi, which is based on the work by Blondel, Guillaume, Lambiotte and Lefebvre [23]. It distinguishes clusters as groups of actors that are densely connected to one another while sparsely connected to other clusters. In other words, the algorithm searches for sparseness within the overall network. In total, *twelve* clusters were found of which the members are grouped accordingly in Fig. 2. Observable is the difference in cluster sizes. There are four clusters that consist of two complementors interacting with each other, but not with the other actors in the ecosystem. The small clusters consist of companies headquartered in the same country. *Sateraito* and *topgate.co.jp*, for example, are both Japanese whereas *Bittle SAS* and *FITNET APPLICATION* are French. The larger clusters reflect more differences. The *Salesforce.com cluster* for instance has a hub-and-spoke network topology, due to the platform efforts of *Salesforce*. Similarly, companies from different geographic locations are grouped around *Zoho Corporation*, making the clusters of *Salesforce.com* and *Zoho Corporation* examples of technology clusters. Other clusters, such as the one in the upper left corner with *OrangeScape* in the middle, lack one central actor and seem to be merely connected to one another through direct and indirect relationships. Geographic location herein does not seem to be an influential factor, as the *OrangeScape cluster* includes multinationals and companies headquartered in the United States, Canada, Mexico, Sweden and the Netherlands.

Noteworthy is the absence of a couple of large actors in Fig. 2. *SaaSt* for instance at present offers eleven applications for Google Apps, yet they are not connected to any other complementor. Similarly, *myERP* is absent while it has nine applications in the app store. Other large complementors such as *Zoho Corporation* (15 applications), *Cloud Technology Solutions* (14 applications) and *TopSolutuions* (10 applications) are among the most central actors in the ecosystem.

V. Analysis

This section describes the analysis of the Google Apps ecosystem and answers the research question. The remainder of this section distinguishes between a quantitative and qualitative analysis. The quantitative part of the analysis answers the research question based on the descriptives summarized in the preceding section. The qualitative part of the analysis considers the input received from the complementors that were contacted by email. Out of the thirty-five complementors, *ten (29%)* returned an answer. Respondents were equally spread over the two contacted categories, meaning that both well embedded actors as active developers responded, providing a more representative input.

A. Analysis of Quantitative Data

Relationship establishment among complementors makes up a critical part of the success of an ecosystem. Prominent actors in the ecosystem seek to extend their relationship portfolio as it eases access to resources and their means to exert power in the ecosystem [11], [5]. Similarly, small complementors seek partnerships under pressure of increasing specialization and technological integration with existing complements in the ecosystem to benefit from established reputation [5]. Moreover, the platform owner benefits from interconnectivity, since it indicates commitment to the platform [7]. Also, increasing embeddedness of actors increases lock-ins, as connected elements are less likely to depart the ecosystem [7]. Subsequently, the argument is that increasing application development and growing interest in relationship establishment coincide, due to the increased power position of central players [9], [5], [10]. To test this argument, the following hypothesis is formulated:

H1: There is a positive relationship between the number of applications an actor develops and its embeddedness in the ecosystem.

The hypothesis is tested by means of a Pearson correlation analysis. For every complementor, the number of applications they developed were considered as the first variable, and the number of partners (i.e. embeddedness) as the second variable. As shown in the SPSS output fragment in Fig. 3, the two variables significantly correlate *(0.232)*, meaning that complementors that develop more applications engage in more formal relationships and the growth in the two variables thus coincides. The correlation is only significant at the *0.01* level since large complementors such as *SaaSt* and *myERP* have not yet established any relationships. A likely explanation for this comes from one of the respondents, who indicated to *"not have established relationships yet, as the focus of a new entrant lies on the extension of its product portfolio"*. Also, new entrants may have partnerships in before their actual entrance to the ecosystem, which has a mitigating effect on the correlation.

Correlations

		NumberOfApplications	Degree
NumberOfApplications	Pearson Correlation	1	,232**
	Sig. (2-tailed)		,000
	N	992	992
Degree	Pearson Correlation	,232**	1
	Sig. (2-tailed)	,000	
	N	992	992

**. Correlation is significant at the 0.01 level (2-tailed).

Fig. 3. SPSS output for correlation between number of applications developed and embeddedness

B. Analysis of Qualitative Data

During the survey among complementors, companies were presented with a description of the Google Apps ecosystem as described in the previous section. When asked, *90%* indicated that the information collected about their company and its relationships is accurate and complete. One respondent indicated to have additional relationships with Google Apps vendors, but omitted to list these engagements on their website. Few respondents indicated to be aware of the network topology of the ecosystem or their own structural position within this network. Most complementors indicated to have an overview of their own relationship portfolio, but lack insight into the dynamics of the network as a whole.

The stance towards relationship management differs among participants in the ecosystem. The majority of respondents indicated to be selective in their partnering activities, limiting themselves to key partnerships. The companies indicated to seek for technological partnerships and integrations with other application vendors to *"complement existing applications"*, evidencing the finding that relationship initiation in the Google Apps ecosystem is driven by technological needs rather than geography. Accordingly, matchmaking among complementors in the ecosystem could prove more effective when the process is based on technological needs. Other vendors, however, indicated to not see any added-value with regard to relationship initiation within the ecosystem, confirming earlier findings of case studies performed with software vendors [24].

VI. Discussion

The results presented in this paper were based on empirical data, obtained by automated and manual data collection from the Google Apps Marketplace. Since the data collection process is labor intensive, to the knowledge of the authors it is the first time a similar approach has been used to visualize the ecosystem around a proprietary platform. Accordingly, a validation was implemented to evaluate the completeness and accuracy of the data collection method. Results of the validation showed that the method was found accurate.

Regardless of the results of the validation step, reservations have to be made with regard to data completeness, due to usage of proprietary sources. For example, some complementors indicated on their website to have technological partners or to employ partnership models, while they omitted to list the companies that constitute their relationship portfolio. To overcome this limitation, CrunchBase was used to identify key relationships that may have been absent on the website.

VII. Conclusion

This paper described the network topology of the Google Apps ecosystem. The analysis investigated the influence that the number of applications developed by an actor has on its embeddedness (i.e. its number of partners), as the hypothesis was that increasing development activity coincides with a growing emergence to establish relationships. By means of app store data extraction metadata about vendors was collected. After that, formal relationships (e.g. technological partnerships, alliances) in the ecosystem were manually identified by creating partner lists for every vendor, based on the information available on company websites and CrunchBase.

At present, the ecosystem is populated by 992 complementors that develop 1341 complementary applications for Google Apps. Illustrative for the recent growth of the ecosystem,is the number of vendors that develop one complement (83%). The ecosystem is a sparsely connected network with on average 1.26 relationships per actor, as 73% of the complementors is solely connected to Google. The ecosystem is composed of twelve clusters, of which larger clusters are technology-based while the small clusters were found to consist of geographic partnerships. When asked, complementors indicated to not be aware of the current topology of the Google Apps ecosystem, and to select their partners based on technological complementarity. To provide an answer to the research question, a correlation analysis was performed, returning a significant correlation of 0.232 evidencing a positive relationship between the number of applications developed by a complementor and its need to establish relationships.

This paper provides a step towards better understanding of the mechanisms at play in proprietary platform ecosystems. Longitudinal studies are an important direction for further research, as they provide insight in the evolution of the ecosystem over time, as well as the influence of changing strategies of individual companies on the ecosystem as a whole. As this research focused on software products, more research is needed to identify the role of service providers or system integrators for value creation within the ecosystem. Furthermore, this ongoing research project aims to compare multiple proprietary platform ecosystems, to uncover the factors shaping the network topology of the ecosystem.

References

[1] A. Gawer, "Platform dynamics and strategies: From products to services," in *Platforms, Markets and Innovation*, A. Gawer, Ed. Cheltenham, UK: Edward Elgar Publishing, 2009, pp. 45–76.

[2] S. Hyrynsalmi, A. Suominen, T. Makila, A. Jarvi, and T. Knuutila, "Revenue Models of Application Developers in Android Market Ecosystem," *Proceedings of the Third International Conference on Software Business*, pp. 209–222, 2012.

[3] C. Baldwin and C. Woodard, "The architecture of platforms: A unified view," in *Platforms, Markets and Innovation*, A. Gawer, Ed. Cheltenham, UK: Edward Elgar Publishing, 2009, pp. 19–45.

[4] G. Briscoe, S. Sadedin, and G. Paperin, "[biology of applied digital ecosystems]," *Proceedings of the First IEEE International Conference on Digital EcoSystems and Technologies*, pp. 458–463, 2007.

[5] M. Iansiti and R. Levien, "Strategy as Ecology," *Harvard Business Review*, vol. 82(3), pp. 68–78, 2004.

[6] L. Uden and E. Damiani, "The future of E-learning: E-learning ecosystem," *Proceedings of the First IEEE Digital EcoSystems and Technologieis Conference*, pp. 113–117, 2007.

[7] A. Gawer and M. Cusumano, "How companies become platform leaders," *MIT Sloan Management Review*, vol. 49, no. 2, pp. 28–35, 2008.

[8] S. Jansen, S. Brinkkemper, and A. Finkelstein, "Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems," *Proceedings of the First International Workshop on Software Ecosystems*, pp. 34–48, 2009.

[9] R. Hanneman and M. Riddle, *Introduction to social network methods*. Riverside, CA: University of California, 2005.

[10] J. Scott, *Social Network Analysis: A Handbook*, 2nd ed. Gateshead, UK: Sage Publications, Inc., 2000.

[11] B. Iyer, C. Lee, and N. Venkatraman, "Managing in a Small World Ecosystem: Some Lessons from the Software Sector," *California Management Review*, vol. 48, no. 3, pp. 28–47, 2006.

[12] J. West, "How Open is Open Enough? Melding Proprietary and Open Source Platform Strategies," *Research Policy*, vol. 32, no. 7, pp. 1259–1285, 2003.

[13] K. Boudreau, "Open Platform Strategies and Innovation: Granting Access vs. Devolving Control," *Management Science*, vol. 56, no. 10, pp. 1849–1872, 2010.

[14] R. Perrons, "The open kimono: How Intel balances trust and power to maintain platform leadership," *Research Policy*, vol. 38, no. 8, pp. 1300–1312, 2009.

[15] C. Burkard, T. Draisbach, T. Widjaja, and P. Buxmann, "Software ecosystems: Vendor-sided characteristics of online marketplaces," *Proceedings of the 41st Annual Conference of the Gesellschaft fur Informatik*, 2011.

[16] M. Harman, Y. Jia, and Y. Zhang, "App Store Mining and Analysis: MSR for App Stores," *Proceedings of the Nineth Working Conference on Mining Software Repositories*, pp. 108–111, 2012.

[17] A. Quaadgras, "Who joins the platform? the case of the RFID business ecosystem," *Proceedings of the 38th Hawaii International Conference on System Sciences*, pp. 269–278, 2005.

[18] R. Basole, "Visualization of interfirm relations in a converging mobile ecosystem," *Journal of Information Technology*, vol. 24, pp. 144–159, 2009.

[19] M. Bengtsson and S. Kock, "'Coopetition' in Business Networks – to Cooperate and Compete Simultaneously," *Industrial Marketing Management*, vol. 29, no. 5, pp. 411–426, 2000.

[20] R. K. Yin, *Case Study Research: Design and Methods*. Thousand Oaks, CA, USA: Sage Publications, Inc, 2009.

[21] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An Open Source Software for Exploring and Manipulating Networks," *Proceedings of the Third International Conference on Weblogs and Social Media*, 2009.

[22] P. Bonacich, "Some unique properties of eigenvector centrality," *Social Networks*, vol. 29, no. 4, pp. 555–564, 2007.

[23] V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics*, vol. 5, no. 10, 2008.

[24] J. van Angeren, V. Blijleven, and S. Jansen, "Relationship Intimacy in Software Ecosystems: A Survey of the Dutch Software Industry," *Proceedings of the International Conference on Management of Emergent Digital EcoSystems (MEDES'11)*, pp. 68–75, 2011.

Exploring the Relationship between Platform Strategy and Interfirm Network Structure: An Analysis of the Office365 Ecosystem

Joey van Angeren, Slinger Jansen, and Sjaak Brinkkemper

Department of Information and Computing Sciences, Utrecht University
Princetonplein 5, 3508 TB Utrecht, the Netherlands
{J.vanAngeren,Slinger.Jansen,S.Brinkkemper}@uu.nl

Abstract. Platform owners face complex decisions in managing a platform ecosystem populated by third-party application developers. At present, little is known about the effect a platform strategy has on the productivity, or interaction among complementors in the ecosystem. This paper presents an exploration of the Microsoft Office365 ecosystem by means of network analysis and bivariate statistics, to uncover the extent to which active complementor retention influences the network structure of the ecosystem. Results show the Office365 ecosystem is populated by 550 complementors, that developed 1204 applications and initiated 787 interfirm relationships. Statistical inference reveals that increased development activity and the status of a complementor as partner coincide with increased network density, implying that complementor locking results in a more cohesive ecosystem. Yet, productivity remained unaffected by regulations around the platform. Results presented in this paper provide increased understanding of the influence of platform strategies, to the benefit of practitioners and academia.

Key words: app store, industry platform, lock-in, network analysis, platform strategy, software ecosystem

1 Introduction

The number of industry platforms and app stores in the software industry is rising. An industry platform is defined by Gawer [1] 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." While initially thought of as the epitome of the mobile era [2], recently Windows 8, SAP BusinessOne, Google Apps, Microsoft Office365 and others started to thrive on third-party application development. In similar environments indirect network effects are prevalent, meaning that an increase in available complementary applications will increase the number of users of the platform [3]. Accordingly, the platform owner becomes dependent on an ecosystem populated by the developers of software products and the interfirm relationships among them [1, 4].

Platform owners have a plethora of instruments at their disposal to govern their platform and the ecosystem around it. In studying these instruments, prior research predominantly focused on their outcomes in market settings [5, 6], whereas the impact on the ecosystem of complementors remained unexplored. Addressing this deficiency, this paper examines the network structure of the Microsoft Office365 ecosystem. It explores the influence of commonly applied [1, 5] active retention and selection of complementors on the network density of the ecosystem. The study regards increasing development activity of a complementor or its status of Microsfot partner as invoked by complementor locking, and assesses the influence thereof on the number of initiated interfirm relationships. As such, this paper answers the following research question: *"What is the influence of complementor lock-ins on the network structure of a proprietary platform ecosystem?"*

This paper presents a visualization and analysis of the ecosystem around the business platform Office365 by means of network analysis [7, 8] and subsequent bivariate statistical interference. It makes use of data collected from the Office365 app store (Office365 Marketplace) and directly from the websites of complementors. Through the analysis of the Office365 ecosystem, this research gains insight into the factors that shape proprietary platform (i.e. a platform that is closed source and owned by a single for-profit entity) ecosystems. In addition, this research adds to the growing body of research on industry platforms and the multitude of platform strategies that exists.

The remainder of this paper continues with an overview of related literature and network metrics in Section 2. An elaboration upon the research approach is included in Section 3. Section 4 presents a description of the Office365 ecosystem and its population. Section 5 analyzes the Office365 ecosystem and among others evidences that complementor locking is positively related to the network density of the ecosystem, while the development scope of complementors remains unaffected. A discussion of validity is presented in Section 6, followed by a summary of findings and suggestions for future research in Section 7.

2 Background

The amount of literature on multi-sided platforms, ecosystems and network analysis is extensive, yet application of their theories in synergy is limited. This section presents a brief review of contemporary literature on multi-sided platforms and the management thereof. Simultaneously, network metrics are identified and their computation is formalized. Herein, parallels with their application in existing literature are drawn whenever possible.

Platform owners face complex and far reaching decisions in the management of their platform and its surrounding ecosystem. According to West [9] platform owners face the challenge of reaping benefits from a proprietary technology, while the success of this technology is determined by the extent to which complementors and users adopt the platform. Recurrently, platform owners have to strike a balance between open and closed platform strategies [6]. While most prior research has focused on stimulating a continuous influx of new complementors in the ecosystem [6, 4], recent advances have led to a stronger emphasis on retention of existing complementors and not provoking unnecessary competition. Elaborated mechanisms include partnership models [10], enforcing platform exclusivity for applications [11], managing diversity of available applications [12] and facilitating interactions among members of the ecosystem [5]. Active retention in turn is associated with an increase in productivity and a denser ecosystem, because complementors further commit, specialize and integrate [13, 11, 5, 4, 14].

Graph theoretical mathematics developed a plethora of means to address the quantitative inquiry of structural properties of networks and their evolution [7, 8]. Several measures such as network density, centrality and clustering coefficients are useful in the analysis of ecosystem structure. Accordingly, the remainder of this subsection defines and formalizes these measurements as well as illustrates their application in ecosystem analysis, for the purpose of replication and uniformity.

Network density reflects the ratio of the number of relationships that are present in an ecosystem compared to the number of relationships that can theoretically be initiated [8]. As such, it is an indicator of the degree of interconnectivity and collaboration among the inhabitants of an ecosystem [15]. Densely interwoven interfirm networks have also been labeled as being more robust [16] or specialized [13]. Network density is calculated as

$$\Delta = \frac{2E}{V(V-1)} \tag{1}$$

where E is the number of interfirm relationships in the ecosystem and V represents the number of complementors. The measure can take values between 0 (empty graph) and 1 (fully connected graph).

Degree centrality is a commonly applied measure to evaluate the structural position of complementors in an ecosystem. The structural position of a complementor is important, as a large number of interfirm relationships eases access to complementary knowledge and fosters exchange or innovation [17]. Degree centrality is denoted as the ratio of the number of relationships a complementor has to the number of relationships it could theoretically have [18]. The measure is computed as follows

$$C_i = \frac{\sum_{i=1}^{n} a_{ij}}{E_i - 1} \tag{2}$$

Where a_{ij} denotes a relationship between complementor i and complementor j, which is valued **1** if present and **0** otherwise. In existing literature, degree centrality has been used to typify vendors. Quaadgras [19] argues that firms should take their degree centrality into account in their alliance strategy. In the operationalization of business ecosystem health measurement, Den Hartigh, Tol and Visscher [20] characterize partners with a high degree centrality as 'healthy' compared to their peripheral peers.

Centralization is an indicator for the extent to which an ecosystem is centralized around a node, and as such it reflects the ratio of how central the most central node in the network is compared to the centrality of all other nodes [18, 7, 8]. Platform ecosystems will display high centralization, due to the strong dependence of complementors on the platform owner [14]. However, fluctuations in the centralization score may indicate the presence of other catalysts in the ecosystem. Centralization is computed as follows, where C_{max} is the degree centrality of the platform owner.

$$C = \frac{\sum_{i=1}^{n} C_{max} - C_i}{max\left(\sum_{i=1}^{n} C_{max} - C_i\right)} \tag{3}$$

Clustering coefficient reflects the extent at which the connections of a complementor are also connected to one another [21]. The clustering coefficient of the entire ecosystem is the average of all values across the network. The clustering coefficient indicates the degree to which an ecosystem can be divided into clusters of complementors that are tightly connected. The clustering coefficient of a complementor is computed as

$$CC_i = \frac{2e_i}{K_i(K_i - 1)} \tag{4}$$

where e_i is the number of interfirm relationships among connections of a complementor, and K_i is the number of connections of complementor *i*. Strongly clustered networks have been labeled as innovative [17]. Meanwhile, Iyer, Lee and Venkatraman [13] argue that specialized ecosystems will reflect a greater degree of clustering, as specialization increases the need for interfirm collaboration.

3 Research Approach

In this research, the Office365 complementors, their characteristics and the interfirm relationships among them were subject of study. Office365 is a cloud office suite that bundles scalable versions of Microsoft Office, Microsoft Lync, Microsoft Exchange and Sharepoint for use by enterprises and governmental or educational institutions. Third-party applications are found on a dedicated part of the bigger Microsoft Pinpoint Marketplace[1], and can be listed either globally or in one of the 59 region-specific marketplaces. The app store holds metadata for each application, of which *application identifier*, *application name*, *application category*, *developer name* and *developer website* were considered relevant for this research. Solely applications listed under the category *"Applications"* were considered, excluding professional services and on-premises extensions from the dataset.

[1] http://www.office365.pinpoint.microsoft.com

By means of a web crawler, all application specific metadata was extracted from the app store over sixty iterations (one to retrieve the globally listed applications, and 59 to traverse all region-specific marketplaces). The web crawler combined a set of scripts in Java programming language, and was derived from the crawler developed by Burkard, Draisbach, Widjaja and Buxmann [22]. It has been previously used in our exploratory study of the Google Apps ecosystem [14]. The metadata was collected in two stages: an initial identification and the actual collection of the metadata. During the initial identification, the application identifiers were collected by systematically traversing all application categories up to the point at which no new applications were found. Then, all application specific metadata could be read-in from their publicly accessible information pages. Thereafter, pattern templates were defined to save all data in a central MySQL database.

After the removal of duplicated entries from the database, a list of complementors was compiled by means of an SQL query. Contrary to existing studies on interfirm relationships where proprietary alliance databases are used as central data source [13, 17, 15], the interfirm relationships were obtained directly from the websites of complementors by archiving mentions of partnerships. This approach was preferred to move beyond alliances alone, which are likely to reflect industry-wide collaborations. Interfirm relationships were treated as symmetric ties, and maintained in an adjacancy matrix in UCINET. Relationships were coded binary, meaning that **1** indicates the presence of a relationship, whereas **0** denotes its absence. Furthermore, complementor websites were used to assess the current partner status.

Because of its compatibility with UCINET, Gephi was used to visualize the resulting ecosystem. Its structural properties were computed either with Gephi or directly in UCINET. Apart from network analysis [7, 8], subsequent analysis was performed by means of bivariate statistics that include independent samples T-tests and Pearson correlations.

4 Descriptives of the Office365 Ecosystem

This section provides a description of a snapshot of the Office365 ecosystem that was constructed on *13-02-2013*. The units of analysis are the complementors, their characteristics and the interfirm relationships among them. The remainder of this section first elaborates upon the complementors and their characteristics, followed by a description of the Office365 ecosystem.

4.1 Complementors

The Office365 Marketplaces contained *1204* applications developed by *550* complementors. For an application to be included in the Office365 Marketplace, it has to be subjected to technical compatibility and complementary value requirements, with which Microsoft reserves the right to refuse inclusion of applications that are not of direct added-value to the platform. On average, each complementor develops *2.18* applications with a standard deviation of *1.65*, indicating a narrow development scope. Noteworthy is that Microsoft itself is not involved in the development of complementary applications. The largest complementor in the ecosystem lists *39* applications, while *67%* of complementors develop one application. A complete distribution of these figures is included in Table 1. Important to note is that the distribution of complementors may be influenced by Microsoft listing each extension for a Microsoft Office component as a separate complement. When a complementor, for example, develops an application that can work with Microsoft Word, Excel and Powerpoint, it can be found three times in the Office365 Marketplace. Subsequently, the actual number of unique complements for Office365 may be lower than the number found in this research.

Table 1. Distribution of Office365 complementors based on the number of applications developed

# of applications	# of complementors
39	1
32	2
23	1
22	1
16	1
14	1
13	1
11	3
10	3
9	5
8	4
7	7
6	13
5	10
4	23
3	24
2	82
1	368

Most applications are listed under multiple categories. Available applications include business templates for Microsoft Powerpoint, document management functionality for Microsoft Outlook, integrations with third-party programs and customer relationship management or human resource extensions. Due to the overlap in the categorization used by the app store, a categorization of applications cannot be provided.

To foster lock-ins and increase control over the ecosystem [10], Microsoft orchestrates its Microsoft Certified Partner Network. Partners pay membership fees and adhere to mandatory product and resource certification. In return, they receive benefits that may include marketing benefits, direct contact with representatives at Microsoft and participation in joint partner events. At present, *278 (50.50%)* complementors are certified Microsoft partners.

Every complementor is assumed to engage in an interfirm relationship with Microsoft, as they extend the Office365 platform and include their applications in the app store. In total, the ecosystem is connected by *787* interfirm relationships. Every complementor on average initiated *1.43* relationships with a standard deviation of *11.74* relationships. The most well connected actors are *Dell (eleven applications)* with *37* relationships and *Nintex (seven applications)* with *32* connections that are among the most active complementors in the ecosystem.

4.2 Ecosystem

The Office365 ecosystem consists of complementors (nodes) and the interfirm relationships among them (edges). The structural properties of the ecosystem are summarized in Table 2. As expected, the Office365 ecosystem is centralized and sparsely connected with a network density of *0.5%*. These values are higher than the network density of *0.25%* and centralization of *99.95%* that were recorded in previous work for Google Apps [14]. Despite the limited interconnectivity among inhabitants of the ecosystem the overall clustering coefficient is high, indicating that most complementors that do initiate interfirm relationships intensively interact.

Table 2. Network level descriptives for the Office365 ecosystem

Metric	**Value**
Size	551
Network density	0.00500
Centralization	0.9984
Modularity	0.336
Clustering coefficient	0.773

Table 3 summarizes the node level properties for the Office365 ecosystem. The high standard deviation for degree centrality reveals the presence of a small number of well embedded complementors in the ecosystem. Yet, *70.18%* of complementors are solely connected to Microsoft and therefore found in the periphery. This finding may be partly explained by the presence of established Sharepoint complementors in the Office365 ecosystem, the on premises version of which already came with an extension architecture.

Table 3. Network metrics for the Office365 ecosystem

Metric	**Min.**	**Max.**	**Avg.**	**Std. dev.**
Degree centrality	0.00183	1	0.00519	0.0427
Clustering coefficient	0.00215	1	0.773	0.228

A network visualization could be created by moving beyond the hub-and-spoke network topology. Accordingly, the dataset was 'cleansed' by removing Microsoft, and the complementors solely connected to Microsoft. Figure 1 shows the resulting network, in which node sizes are proportional to the number of applications developed per complementor. The colors and grouping represent clusters in the ecosystem, identified by means of the modularity algorithm [23]. The algorithm treats clusters as groups of complementors that are densely connected to one another while sparely linked to the rest of the ecosystem.

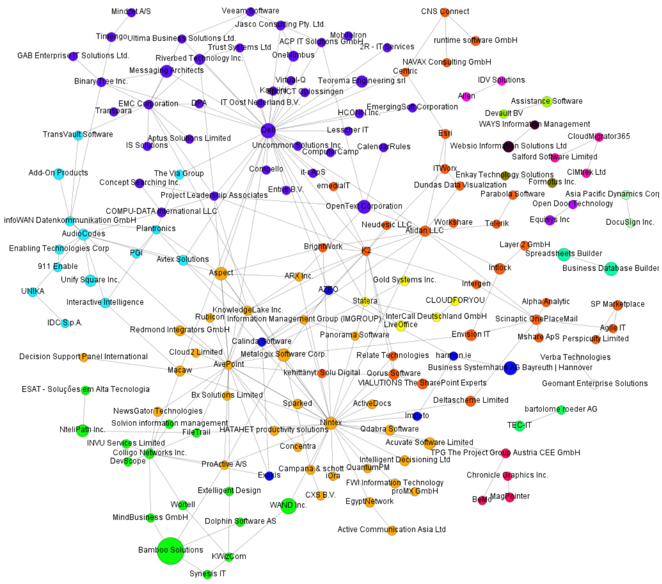

Fig. 1. Network visualization of clusters in the Office365 ecosystem

Figure 1 visualizes the interfirm relationships among *164* complementors. Noteworthy is that most active developers are present. Closer inspection reveals that absent complementors are individuals rather than enterprises. Individual developer *Orlando's VBA and Excel Site*, for instance listed *32* applications in the Office365 app store. The clusters in the ecosystem appear to be well and densely interwoven, apart from the dyads shown in the right of Figure 1. Interfirm relationships seem to pertain technological partnerships, as most clusters appear to lack a geographical focus.

5 Analysis

The increasing involvement of a complementor may well reflect in its emergence to initiate interfirm relationships. As the complmentor starts developing more applications, it will search for interfirm relationships to ease its access to resources or foster its influential position in the ecosystem [8, 13, 11]. Meanwhile, new entrants seek for technological integrations with existing applications to increase their visibility in the market [16, 14]. To test whether this argument holds, a Pearson correlation analysis is performed. The number of applications developed by a complementor is treated as the independent variable, and the number of interfirm relationships it initiated as the dependent variable. As shown in Figure 2, there is a significant positive correlation *(0.131)*, meaning that growth in both variables coincides. Explanations for the mild correlation that is only significant at the *0.01 level* can be sought in the amount of entrants that already established interfirm relationships in before joining the ecosystem. Nevertheless, empirical evidence postulates that the density of an ecosystem serves as a good indicator of the involvement of complementors around a platform.

Correlations

		Complements	Relationships
Complements	Pearson Correlation	1	,131**
	Sig. (2-tailed)		,002
	N	550	550
Relationships	Pearson Correlation	,131**	1
	Sig. (2-tailed)	,002	
	N	550	550

**. Correlation is significant at the 0.01 level (2-tailed).

Fig. 2. SPSS output for Pearson correlation between number of applications developed and number of interfirm relationships per complementor

One can argue that the active partner enablement strategy of Microsoft has a profound impact on the productivity and robustness of Microsoft platform ecosystems. By means of their partnership model, Microsoft can foster lock-in effects, enforce platform exclusivity for complements [1] and further increase (quality) control through certification [10]. To examine whether partners are more active in developing applications and initiating interfirm relationships compared to non partners, an independent sample T-test is performed. Office365 complementors are distributed in two groups based on their current partner status. Partner status is coded as a dummy variable, in which **1** corresponds to a status as Microsoft partner and **0** represents non partners. In total, *278* complementors are acknowledged as Microsoft partners and *272* are categorized as non partners. The independent sample T-test was performed at the *95% confidence interval*. A Microsoft partner in the Office365 ecosystem on average initiated *1.192* interfirm relationships with a standard deviation of *2.925*, a non partner on average had *0.5221* ties with a standard deviation equal to *2.509*. The partner category has a more skewed distribution of interfirm relationships compared to the non partners category. Based on these figures, there is a significant difference in initiated interfirm relationships between partners and non partners; t(538) = 2.895, p = 0.004. These results suggest that Microsoft partners in the Office365 ecosystem have significantly more interfirm relationships compared to non partners and that lock-ins thus reflect in the network density of the ecosystem.

The second independent sample T-test is performed to investigate the relationship between partnership model participation and development activity. Microsoft partners on average develop *2.313* applications with a standard deviation of *3.351*, and non partners develop *2.063* applications with standard deviation of *3.390*. Based on these group means, there is no significant difference in development activity between partners and non partners; t(548) = 0.871, p = 0.384. Results imply that Microsoft partners are not significantly more committed to development of complementarities compared to non partners, the small difference in group means may be attributed to random variation or luck.

In summary, empirical evidence postulates that complementor locking is positively related to the network density of an ecosystem. Meanwhile, the scope of complementors in their development activity appears to remain narrow and limitedly affected by the regulations imposed around a platform. The latter was also argued by Boudrea [12], who hypothesized that application developers are unlikely to move beyond their scope.

6 Discussion

The results presented in this paper were based on data collected from the Office365 app store and directly from websites of complementors. Despite all precautions taken, like any exploratory research this study has a number of limitations. Inherent in the problem domain is the strong reliance on proprietary sources. The lack of transparency becomes evident when complementors indicate to engage in technological partnerships or alliances, but omit to list their actual partners. Accordingly, the partner activity of such a company is neglected, while in practice it might be involved in numerous interirm relationships. To limit the influence of this effect, interfirm relationships were treated as symmetric ties to increase coverage. Furthermore, a preliminary validation of the data collection method in previous research [14] brought it forward as being accurate.

The use of static analysis for networks that are likely to change over time, pose another limitation for this research. While this threat needs to be acknowledged, previous research [13] showed that the network structure of multiple major software ecosystems remained remarkably stable over a 12-year period. Moreover, a remark needs to be placed related to the scope of interfirm relationships, which is likely to span wider than the boundaries of a platform ecosystem. Accordingly, company websites rather than alliance databases were used to include partnerships with a finer level of granularity.

7 Conclusion

This paper presented a description of the Office365 ecosystem. By means of app store data extraction metadata about complementors was retrieved, after which the interfirm relationships among them were manually identified through the creation of partner lists based on partner mentions on complementor websites. Subsequent analysis was performed by means of network analysis and bivariate statistics and oriented itself at investigating the influence of fostering complementor lock-ins on the network structure of a proprietary platform ecosystem.

The Office365 ecosystem was populated by 550 complementors that together developed 1204 applications. Complementors initiated 787 interfirm relationships, which amounted for an average of 1.43 connections per complementor. While 70.18% of complementors was solely connected to Microsoft, the remainder of the ecosystem appears densely connected. Statistical inference evidenced a positive relationship between the development activity of a complementor and the number of interfirm relationships it initiated. In addition, Microsoft partners were found to initiate significantly more interfirm relationships than not partners, yet the number of applications they developed appeared equal. These findings postulated that complementor locking will be positively related to the network density of an ecosystem, while the development scope of complementors remains unaffected and narrow.

This paper provides a step towards better understanding of the effects of enforcing platform regulations by examining the influence of fostering lock-ins on the network structure of a platform ecosystem. Future research should further address the tension between the decisions of a platform owner and their influence on the productivity and embeddedness on complementors. The method laid out in this paper may function as a blueprint for longitudinal replication. Furthermore, this ongoing research project aims to contrast multiple proprietary platform ecosystems to uncover their characteristic similarities and differences.

References

1. Gawer, A.: Platform dynamics and strategies: From products to services. In Gawer, A., ed.: Platforms, Markets and Innovation, Cheltenham, UK, Edward Elgar Publishing (2009) 45–76
2. Holzer, A., Ondrus, J.: Mobile application market: A developer's perspective. Telematics and Informatics **28**(1) (2011) 22–31
3. Rochet, J., Tirole, J.: Platform Competition in Two-Sided Markets. Journal off the European Economic Assosication **1**(4) (2003) 990–1029
4. Jansen, S., Brinkkemper, S., Finkelstein, A.: Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems. Proceedings of the First International Workshop on Software Ecosystems (2009) 34–48
5. Boudreau, K., Hagiu, A.: Platform rules: Multi-sided platforms as regulators. In Gawer, A., ed.: Platforms, Markets and Innovation. Edward Elgar Publishing, Cheltenham, UK (2009) 163–191

6. Boudreau, K.: Open Platform Strategies and Innovation: Granting Access vs. Devolving Control. Management Science **56**(10) (2010) 1849–1872
7. Scott, J.: Social Network Analysis: A Handbook. 2nd edn. Sage Publications, Inc., Gateshead, UK (2000)
8. Hanneman, R., Riddle, M.: Introduction to social network methods. University of California, Riverside, CA (2005)
9. West, J.: How Open is Open Enough? Melding Proprietary and Open Source Platform Strategies. Research Policy **32**(7) (2003) 1259–1285
10. Van Angeren, J., Kabbedijk, J., Popp, K., Jansen, S.: Managing software ecosystems through partnering. In Jansen, S., Brinkkemper, S., Cusumano, M.A., eds.: Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry. Edward Elgar Publishing, Cheltenham, UK (2013) 85–102
11. Gawer, A., Cusumano, M.: How companies become platform leaders. MIT Sloan Management Review **49**(2) (2008) 28–35
12. Boudreau, K.J.: Let a Thousand Flowers Bloom? An Early Look at Large Numbers of Software App Developers and Patterns of Innovation. Organiation Science **23**(5) (2012) 1409–1427
13. Iyer, B., Lee, C., Venkatraman, N.: Managing in a Small World Ecosystem: Some Lessons from the Software Sector. California Management Review **48**(3) (2006) 28–47
14. Van Angeren, J., Blijleven, V., Jansen, S., Brinkkemper, S.: Complementor Embeddedness in Platform Ecosystems: The Case of Google Apps. Proceedings of the Seventh International Conference on Digital EcoSystems and Technologies (2013) 37–42
15. Basole, R.: Visualization of interfirm relations in a converging mobile ecosystem. Journal of Information Technology **24** (2009) 144–159
16. Iansiti, M., Levien, R.: Strategy as Ecology. Harvard Business Review **82(3)** (2004) 68–78
17. Schilling, M.A., Phelps, C.C.: Interfirm Collaboration in Networks: The Impact of Large-Scale Network Structure on Firm Innovation. Management Science **53**(7) (2007) 1113–1126
18. Freeman, L.: Centrality in social networks conceptual clarification. Social Networks **1**(3) (1979) 215–239
19. Quaadgras, A.: Who joins the platform? the case of the RFID business ecosystem. Proceedings of the 38th Hawaii International Conference on System Sciences (2005) 269–278
20. Den Hartigh, E., Tol, M., Visscher, W.: The health measurement of a business ecosystem. Proceedings of the European Network on Chaos and Complexity Research and Management Practice Meeting (2006)
21. Watts, D., Strogatz, S.: Collective dynamics of small world networks. Nature **393** (1998) 440–442
22. Burkard, C., Draisbach, T., Widjaja, T., Buxmann, P.: Software ecosystems: Vendor-sided characteristics of online marketplaces. Proceedings of the 41st Annual Conference of the Gesellschaft fur Informatik (2011)
23. Blondel, V., Guillaume, J., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics **5**(10) (2008)