Can We Ask You To Collaborate? Analyzing App Developer Relationships in Commercial Platform Ecosystems

Joey van Angeren[a,*], Carina Alves[b], Slinger Jansen[c]

[a]School of Industrial Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
[b]Centro de Informatica, Universidade Federal de Pernambuco, Av. Professor Luís Freire s/n, Cidade Universitária, 50740-540 Recife, Brazil
[c]Department of Information and Computing Sciences, Utrecht University, Princetonplein 5, 3508 TB Utrecht, The Netherlands

Abstract

Previous studies have emphasized the necessity for software platform owners to govern their platform ecosystem in order to create durable opportunities for themselves and the app developers that surround the platform. To date, platform ecosystems have been widely analyzed from the perspective of platform owners, however, how and to what extent app developers collaborate with their peers needs to be investigated further. In this article, we study the interfirm relationships among app developers in commercial platform ecosystems and explore the causes of variation in the network structure of these ecosystems. By means of a comparative study of four commercial platform ecosystems of Google (Google Apps and Google Chrome) and Microsoft (Microsoft Office365 and Internet Explorer), we illustrate substantial variation in the extent to which app developers initiated interfirm relationships. Further, we analyze how the degree of enforced entry barriers to the app store, the use of a partnership model, and the domain of the software platform that underpins the ecosystem affect the properties of these commercial platform ecosystems. We present subsequent explanations as a set of propositions that can be tested in future empirical research.

Keywords:
case study, commercial platform ecosystem, ecosystem governance, interfirm network analysis, software ecosystem, software platform

1. Introduction

In the last decade, software ecosystems have gained increased attention within the software industry (Manikas and Hansen, 2013b). The lens of a software ecosystem shifts the traditional perspective of software engineering, where a single company used to develop and commercialize software systems (Bosch, 2009). In the context of a software ecosystem, companies need to focus on inter-organizational collaborations involving several players such as platform owners, app developers, and customers (Jansen et al., 2010). Managing the multi-faceted relationships among these parties is a key success factor for the healthy evolution of a software ecosystem (Iansiti and Levien, 2004b; den Hartigh et al., 2013).

We adopt the definition of a software ecosystem by Jansen et al. (2010, p. 35), who define the concept 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."* Examples of software ecosystems are manifold, but perhaps most illustrative is the ecosystem that emerged around mobile operating system iOS. Shortly after the launch of the first iPhone in 2008, Apple introduced the Apple App Store as a distribution platform for third party software applications for its new mobile device running on the iOS operating system. Inspired by the merits of open innovation (Chesbrough, 2003), Apple cultivated an ecosystem of app developers. The number of applications in the Apple App Store quickly

*Corresponding Author. Tel.: +31 40 247 3046
Email addresses: j.v.angeren@tue.nl (Joey van Angeren), cfa@cin.ufpe.br (Carina Alves), slinger.jansen@uu.nl (Slinger Jansen)

grew from 500 in 2008 to over 1.4 million applications in 2015[1]. Reaping benefits from ready-to-use extension architectures, substantial software reuse, and existing distribution channels app developers adopted the iOS platform en masse. Other prominent examples of *platform ecosystems* include desktop operating systems (e.g., Windows 8, OS X), web browsers (e.g., Google Chrome, Firefox), and business platforms (e.g., Salesforce.com, Google Apps).

The fast-paced advent of platform ecosystems brings several challenges to their owners. Platform owners have become dependent on the extensions and applications built within their ecosystem to maintain their success, while app developers also depend on the size of the installed base of the software platform to thrive. Albeit that the members of the ecosystem share success, not all members carry equal responsibility for the creation and governance of the network (Iansiti and Levien, 2004b; Gawer and Cusumano, 2008; Boudreau and Hagiu, 2009; Jansen et al., 2012; den Hartigh et al., 2013). Hence, prior empirical research has explored such questions as how and when to open up an ecosystem to increase the involvement of app developers (Jansen et al., 2012), how to maintain persistent software development activity among app developers (Gawer and Cusumano, 2008), and in what ways can a platform owner manage competition among its app developers (Boudreau, 2012). However, underlying many of these issues is a lack of understanding of how – and the extent to which – app developers collaborate, such as through alliances, shared research and development, and less formalized means of interfirm collaboration including mutual product certification and technological partnerships. It is important to understand the ways in which a platform owner can foster collaboration among app developers because the firm directly benefits from co-creation (Gawer and Cusumano, 2008). These questions are particularly relevant for commercial platform ecosystems that, to date, have barely been investigated in previous studies. Notable exceptions include visualizations of alliance networks of IBM, Microsoft, and SAP between 1990 and 2002 (Iyer et al., 2006), visualizations of the mobile and ICT ecosystem (Basole, 2009; Basole and Karla, 2011; Basole et al., 2014), and a qualitative study of the SAP partner ecosystem (Rickmann et al., 2014).

To increase the understanding of governance of commercial platform ecosystems, we explore and compare four ecosystems that emerged around software platforms of Google and Microsoft. In particular, we aim at investigating the relationships among app developers in these ecosystems. We address two research questions:

1. *What are the characteristics of interfirm relationships in commercial platform ecosystems?*
2. *How do governance mechanisms such as entry barriers to the app store, partnership models, and the domain of the underpinning software platform affect the initiation of interfirm relationships among app developers in commercial platform ecosystems?*

We investigate the Google Apps, Microsoft Office365, Google Chrome, and Internet Explorer ecosystems. The first two ecosystems are canonical for an emerging set of business productivity platforms whereas the latter two platforms compete in the web browser domain. We study the ecosystems of Google and Microsoft because both firms adopt distinct governance philosophies – Google and Microsoft embody the traditional tension between *'open'* and *'closed'* strategies in the software industry, respectively. Therefore, the analysis of these ecosystems provides a rich context to explore variations in network structure. We analyze the ecosystems on dimensions such as size, network density, and others. In doing so, we illustrate that there is substantial variation in the network structure of the ecosystems that we studied. We assess the extent to which ecosystems that are governed by the same platform owner exhibit similar structural properties. Further, we compare the structure of ecosystems that are underpinned by comparable software platforms.

Our study aims to make several contributions. The research presented in this article builds on a series of studies that aims to investigate the structural properties of commercial platform ecosystems. In particular, our work advances previous studies that explored the structures of the Google Apps (van Angeren et al., 2013a) and Microsoft Office365 (van Angeren et al., 2014) ecosystems in isolation. In this article we examine the differences between these ecosystems that are governed with distinct strategies. We extend the prior exploration of commercial ecosystems by Iyer et al. (2006). The authors call for a 'networked scorecard' to evaluate how managerial decisions can impact the ecosystem at large, but such a method lacks practical applicability without a proper understanding of the factors that underlie interfirm network structure. This article also provides two key contributions to the software ecosystems field.

[1]http://www.apple.com/pr/library/2015/01/08App-Store-Rings-in-2015-with-New-Records.html (accessed May 8, 2015)

First, we provide an in-depth analysis of interfirm relationships within commercial platform ecosystems, a perspective that to date has remained deficient. Second, we compare the network structures of the four studied commercial platform ecosystems to understand what factors affect the initiation of interfirm relationships among app developers. In their systematic literature review, Manikas and Hansen (2013b) signal that the vast majority of studies focus on open source ecosystems, thereby neglecting the distinct characteristics of commercial ecosystems. In addition, previous studies have mainly focused on the single perspective of platform owners (e.g., Gawer and Cusumano, 2008; Eisenmann et al., 2009; Boudreau, 2012; Ghazawneh and Henfridsson, 2013; Jansen and Cusumano, 2013). Our research complements this line of research by taking a more holistic perspective of app developer relationships within commercial software ecosystems.

The remainder of this article continues with an outline of the extant literature relevant to our study in Section 2. Section 3 outlines our research method. It describes the context of the platform ecosystems of Google and Microsoft that we studied, and it explains how we extracted and analyzed data. In Section 4, we describe each commercial platform ecosystem by providing elementary descriptives and visualizing the interfirm relationships among principal app developers. Section 5 presents a comparison among the four studied ecosystems. This comparison is followed by Section 6 in which we outline both theoretical and practical contributions, and limitations of our study. Finally, we summarize our main findings and provide directions for future research in Section 7.

2. Background

The interconnectivity of the software industry has increased dramatically over the past few decades. Product development has moved from the internals of an organization through supply chains and software product lines to software platforms that are now omnipresent in various segments of the industry (Gawer, 2009). Simultaneously, software ecosystems emerged as the software industry-specific lens of the business ecosystem concept (Moore, 1993). Software ecosystems research is largely interdisciplinary, it encompasses studies from software engineering, innovation, and management (Manikas and Hansen, 2013b). In its most simple form, a software ecosystem involves groups of actors that collaborate around a common technology, such as a software product line or a software platform (Hanssen, 2012).

Presumably fueled by the widespread availability of software repositories (Kagdi et al., 2007), extant empirical research on collaboration in software ecosystems has mostly focused on open source communities (Manikas and Hansen, 2013b). For example, Kabbedijk and Jansen (2011) visualized the relationships among developers of the Ruby on Rails community and found that much of the coordination effort within the ecosystem was carried out by a limited number of committed developers. Moreover, the authors found that approximately 90 percent of the activity in the ecosystem was generated by 10 percent of the ecosystem members. In similar vein, a number of studies illustrates the great degree of decentrality in open source ecosystems (e.g., Madey et al., 2002; Crowston and Howison, 2005; Grewal et al., 2006; Lungu et al., 2010). Madey et al. (2002) studied open source projects that were maintained in the SourceForge repository. The authors argued that two open source developers are related if they contributed to the same open source project. Madey and colleagues found that most developers were involved in a limited number of projects, mutually connected to the greater developer community through a couple of developers that contributed to many projects at the same time (i.e., 'linchpins'). Contrary to Madey et al. (2002), Lungu et al. (2010) investigated both project (i.e., technical) dependencies and developer (i.e., collaboration) interdependencies in open source ecosystems. By means of an exploratory case study, the authors illustrated that around half of developers were not connected to any other developer, yet some of their projects were principal to the software development in the ecosystem.

Contrary to open source ecosystems, commercial ecosystems are much more centralized. In a commercial platform ecosystem, the owner of the technology is referred to as the 'keystone', or 'reference organization'. From a more networked point of view, the keystone has also been attributed the role of a 'hub' because it interacts with all other members of the ecosystem (Iyer et al., 2006; Burkard et al., 2012; Kude et al., 2012). The platform owner provides the architecture of the platform and its core functionality, and it is surrounded by app developers (i.e., niche players or spokes) that build extensions and applications to complement the platform. As such, the ecosystem can be visualized as a hub-and-spoke network in which a platform owner is surrounded by a large number of app developers (Chellappa and Saraf, 2010; Kazman et al., 2012; Kude et al., 2012; van Angeren et al., 2013a, 2014). Hub-and-spoke networks such as commercial platform ecosystems can be regarded as 'loosely coupled systems' (Orton and Weick, 1990). A platform owner and its app developers are generally not linked through well-defined interfirm relationships such as joint ventures or strategic alliances. Rather, the app developers collaborate and co-create through more informal interfirm relationships that include certification of the other party's product, technological collaboration, and joining partnership models (Ceccagnoli et al., 2011; Kude et al., 2012; van Angeren et al., 2013b).

In a commercial platform ecosystem, the platform owner is the prime responsible for orchestration, coordination, and governance (Iansiti and Levien, 2004a,b). The platform owner strives for a continuous increase in the installed base of the platform, while at the same time aiming to maximize its own profits (Jansen and Cusumano, 2013). In the presence of so-called network externalities (Katz and Shapiro, 1985) – which means that the adoption of the platform by consumers is dependent on the amount of compatible software products developed by app developers and vice versa – ecosystem governance becomes a critical issue. The platform owner has to provide durable opportunities for prospective app developers, which may adversely affect its own profits (West, 2003). Ecosystem governance involves the use of strategic procedures and processes to control, maintain, or change the ecosystem (Jansen and Cusumano, 2013). It encompasses both technical and business aspects that include the management of the software platform and its interfaces, definition of a viable business model, formulation of entry barriers, and partner development. The efficacy of ecosystem governance mechanisms can be assessed in terms of the health of the software ecosystem (Iansiti and Levien, 2004b; den Hartigh et al., 2013); a set of indicators that reflects its longevity and propensity for growth.

In an attempt to better inform platform owners in the strategic management of their commercial platform ecosystem, a nascent stream of literature (e.g., Gawer and Cusumano, 2008; Jansen et al., 2012; Jansen and Cusumano, 2013; Rickmann et al., 2014) has started inventorying the plethora of governance mechanisms that exist. Two often cited examples of such mechanisms include the definition of entry barriers and the creation of a partnership model. With the formulation of entry barriers, a platform owner determines how easy or difficult it is for prospective app developers to join the ecosystem (Boudreau, 2010; Jansen and Cusumano, 2013). Conversely, a partnership model provides app developers that are already part of the ecosystem with the incentives or resources to foster their growth (van Angeren et al., 2013b; den Hartigh et al., 2013). Participants in the SAP partnership model for instance benefit from access to an extensive database with customers, the Eclipse Foundation uses its partnership model to foster co-creation by matching partners who have similar interests, and Microsoft provides its partners with a plethora of marketing materials and organizes annual developer conferences.

Albeit that the utility an app developer gets from joining a platform ecosystem is well established – participating in a platform ecosystem has shown to positively affect an app developer's sales as a consequence of network externalities and economies of scale (Ceccagnoli et al., 2011) – to date the exact efficacy of governance mechanisms has remained elusive. Extant inquiry in this line of research has mainly directed itself at studying the factors that motivate an app developer to enter an ecosystem. Based on case studies in the enterprise software industry, Kude et al. (2012) posit that four types of resources and capabilities of the platform owner are principal in motivating an app developer to join the ecosystem. The motivations include the ability to provide an integrated platform, the ability to innovate systems, the capability to provide app developers access to broad markets, and the reputation of the software platform. Koch and Kerschbaum (2014) find that, next to the intrinsic motivation of app developers, the governance strategies of a platform owner inform ecosystem selection. Notwithstanding their contributions, neither of the former studies develops hypotheses about the way in which a governance mechanism may affect the structural properties of a commercial platform ecosystems, or the way in which these effects may deviate across ecosystems.

3. Research Method

Given the limited theory available on collaboration in, and governance of, commercial platform ecosystems (cf., Manikas and Hansen, 2013b), we conducted an inductive multiple case study of two ecosystems of Google (Google Apps and Google Chrome) and Microsoft (Microsoft Office365 and Internet Explorer). Inductive case study research is especially appropriate for relatively novel research domains, where there are limited established theories to rely on, or where the phenomenon under study is unclear (Eisenhardt, 1989; Yin, 2009). Our case study selection and multiple case study design allowed us to perform comparative analysis (Eisenhardt, 1991). In the following subsections we present our research context in greater detail, followed by a description of the procedures for data collection and analysis.

3.1. *Study Context: Four Commercial Platform Ecosystems of Google and Microsoft*

We selected ecosystems of Google and Microsoft as case study subjects because both firms adopt canonical and extreme governance philosophies. The ecosystems embody the traditional tension between *'open'* and *'closed'* strategies in the software industry (West, 2003; Jansen et al., 2012). Microsoft is known for molding proprietary standards and raising entry barriers to its ecosystems. Conversely, Google – that for instance made part of the source code of some of its products available to the general public – is perceived to be transparent and open. Microsoft and Google compete in numerous segments of the industry, thus providing a rich context for comparative analysis of network structures of commercial platform ecosystems. It provides an avenue to compare both ecosystems that exist around similar underpinning software platforms, and ecosystems that are governed by the same platform owner. Put differently, our research design allowed us to explicitly compare ecosystems along two critical dimensions: the governance philosophy by which the ecosystem is managed, and the domain of the software platform that underpins the ecosystem. The comparative analysis was performed along the axes of a two-by-two matrix, which is illustrated in Figure 1. Our case study subjects represent two distinct yet prominent segments of the software industry. Google Apps and Microsoft Office365 are direct competitors in the enterprise productivity market, whereas Google Chrome and Internet Explorer are illustrative examples of business-to-consumer products that gradually evolved into platform ecosystems (Gawer and Cusumano, 2008).

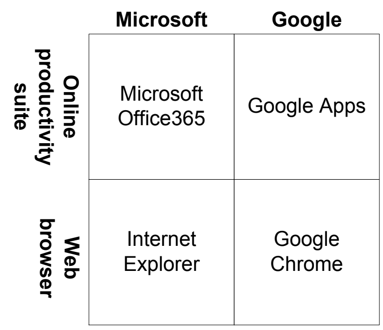

Figure 1: Classification of the commercial platform ecosystems studied

Google Apps[2] is Google's cloud-based productivity suite for small to medium-sized enterprises and educational institutions. The platform was launched in 2006 and consists of Google products such as Gmail, Google Calendar, Google Drive, Google Sites, and others. App developers extend the platform to integrate it with complementary suites, introduce customer relationship management functionality, or provide cloud migration services. Google distributes third party applications through the Google Apps Marketplace.

Google Chrome[3] is the web browser launched by Google in 2008. The platform is partially open source and therefore more accessible for app developers. Google Chrome came with an extension architecture at its initial launch. Third party applications can be found on the Chrome Web Store. Examples of available third party applications include games, web developer tools, ad blockers, and tools for password management.

Microsoft Office365[4] is Microsoft's equivalent to Google Apps that was first released in 2010. The platform is intended for small to medium-sized enterprises and governmental or educational institutions. The platform consists of customizable versions of Microsoft Exchange, Microsoft Lync, Microsoft Office Online, and Microsoft SharePoint. Although Microsoft does not provide an extension architecture for the platform as a whole, app developers can separately extend the functionality of each of the platform components. These third party applications are included in the global Microsoft Pinpoint marketplace or one of its 59 region-specific counterparts. Third party applications include varieties such as integrations, add-ons, business templates for Microsoft Office, and others.

[2]http://www.google.com/enterprise/marketplace
[3]http://chrome.google.com/webstore
[4]http://office365.pinpoint.microsoft.com

Internet Explorer[5] is the longstanding web browser of Microsoft that was introduced in 1995. Its third party applications are presented in the Internet Explorer Gallery, from where they can be retrieved and installed. Examples of available third party applications for Internet Explorer include parental control add-ons, toolbars, and accelerators.

3.2. Data Collection

Principal to any network analysis is the definition of boundaries of the network under study (Scott, 2000). Consistent with prior studies (cf., Iyer et al., 2006; Basole and Karla, 2011; Burkard et al., 2012; van Angeren et al., 2013a, 2014), we defined the members of an ecosystem as: *those companies, individuals, and developer communities that sell or distribute one or more applications through the official app store of the platform*. From this group of app developers, our study investigated those that offered software-as-a-service applications. We excluded app developers that offered professional services and on-premises applications. It follows from our definition of ecosystem membership that we consider those interfirm relationships that are initiated between two app developers are both members of the same ecosystem.

In order to ensure comparability of network structures across platform ecosystems of Google and Microsoft, we collected data for all four case studies in accordance with a predefined protocol (Eisenhardt, 1989; Yin, 2009). We performed our data collection according to the following phases: (1) exploration of ecosystem governance, (2) identification of app developers, and (3) mapping of interfirm relationships. These procedures were followed by a survey with a subset of app developers from the Google Apps ecosystem. Our data collection started in February 2013 and lasted around four months. The remainder of this subsection presents our data collection procedures in greater detail.

Phase 1 (exploration of ecosystem governance): to observe the entry requirements to be met by prospective app developers and the partnership and certification programs in place, we analyzed the documentation available about the studied platforms. Documents of interest included information for prospective ecosystem members, architectural documentation, certification criteria, and partnership model documentation. All relevant documents were stored in a central database for later reference.

Phase 2 (identification of app developers): we developed a web crawler to identify the app developers that had one or more applications in the app store of the four studied platforms at the time that the data was collected. The web crawler combined a set of scripts for data extraction from the Internet, and was partly based on prior work by Burkard et al. (2012). Data collection started by identifying all available applications and the URLs to their information web pages. Depending on the architecture of the app store (i.e., the way in which the app store presents an overview of available applications), a list of all applications was either retrieved from a complete application listing (Google Apps and Google Chrome), per category of applications (Microsoft Office365) or per page in the app store (Internet Explorer). Raw data from individual application information web pages was retrieved and parsed in accordance with predefined pattern templates that describe how specific information is presented in the app store. Albeit that the availability of data differs per app store, we defined pattern templates to at least collect the following data from each app store: *application identifier*, *application name*, *application category*, *app developer name*, *app developer website*, *certification status*, and *partner status*. After removal of duplicates, all data was stored in a central database.

Phase 3 (mapping of interfirm relationships): prior studies that visualized interfirm networks (e.g., Iyer et al., 2006; Rosenkopf and Schilling, 2007; Schilling and Phelps, 2007; Basole, 2009) have relied on proprietary alliance databases to identify interfirm relationships. Commonly used databases include Connexiti, Lexis-Nexis, and the International Data Corporation Platinum Database. Since the scope of our study is limited to the boundaries of commercial platform ecosystems, such databases do not provide information of sufficient granularity. For instance, these databases do not document more informal interfirm relationships that are initiated in the context of the studied ecosystems, such as mutual product certification or technological integration. Therefore, we chose to identify interfirm relationships directly from the websites of app developers. One of the authors examined all app developer websites to identify mentions of partnerships. Examples of such mentions

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

included alliances, technological partnerships, collaborative research and development, strategic partnerships, mutual product certification, technological integration, and partnership model participation, usually presented on website tabs labeled *'Friends'*, *'Partners'*, *'Partnerships'*, *'Strategic Partnerships'*, or *'Collaborators'*. This method was preferred over automated inspection of websites due to the limited level of standardization in company websites. To illustrate, some websites merely included a list of partners, while others outline their partners by presenting company logos or descriptions of partnerships. Further, we used the openly accessible company database CrunchBase[6] to triangulate data on identified interfirm relationships. Interfirm relationships were treated as binary and symmetric ties that were stored in an adjacency matrix[7].

Phase 4 (survey with Google Apps app developers): to increase the reliability of our study as well as to obtain additional insights, we sent a brief email questionnaire to Google Apps app developers. In total, we contacted thirty-five app developers. This group encompassed seventeen app developers that developed the most applications and eighteen app developers that had the largest number of interfirm relationships. Ten app developers (*28* percent) responded to our survey. We presented the app developers with a brief outline of our preliminary findings and an overview of the data we collected regarding their company, which also included a list of applications and identified interfirm relationships. We posed the following questions:

1. Is the list of applications that we compiled for your company accurate and complete?
2. Is the list of interfirm relationships within the Google Apps ecosystem that we compiled for your company accurate and complete?
3. How likely is the initiation of new interfirm relationships between your company and other Google Apps app developers in the near future?

After analyzing the responses that we received, we confirmed that our data collection procedure was effective and reliable. Just one respondent indicated that his company initiated interfirm relationships within the boundaries of the ecosystem that they decided not to mention on their company website. Responses were received from the CEO (seven), head of marketing (two) and business development manager (one).

3.3. Data Analysis

The raw data that was extracted from various sources was tabulated prior to analysis in order to maintain a chain of evidence. Ucinet, a social network analysis software package, was used to obtain network structure measures on each of the ecosystems studied (Borgatti et al., 2002). Gephi, a software package for the visualization of social networks, was used to create graphical images of the ecosystems (Bastian et al., 2009). In each figure, the members of a commercial platform ecosystem are visualized as nodes, and the interfirm relationships among them as edges. To move beyond the dominant hub-and-spoke topology of commercial platform ecosystems app developers only connected to the platform owner, and subsequently the platform owner itself were excluded for the purpose of visualization. The modularity algorithm (Blondel et al., 2008) was used to identify clusters (e.g., groups of ecosystem members that are tightly connected to one another) among the remaining app developers[8].

Table 1 lists the descriptive statistics of the four ecosystems studied. The first four metrics – *entry barriers*, *partnership or certification model*, and *number and percentage of app developers with partner or certification status* respectively – relate to the degree of entry barriers and partner development catered by a platform owner in order to steer its ecosystem. Entry barriers are those requirements that a prospective developer has to meet in order to get its first (or subsequent) app published in the app store of the platform. Common examples of such entry barriers include technological validation of the submitted application and payment of a developer or validation fee. The latter three descriptive statistics reflect whether the platform owner implemented a partnership model or formal product certification program, and if so, what the coverage is across the ecosystem.

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

[7] An adjacency matrix is a square matrix with ecosystem members as rows as columns. Entries in the adjacency matrix, for ecosystem members i and j denoted as a_{ij}, indicate the ecosystem members that are interrelated (i.e., adjacent). In a binary adjacency matrix, a_{ij} takes a value of either 1 (present) or 0 (absent). In a symmetric adjacency matrix, the value of a_{ij} is equal to a_{ji}.

[8] The modularity algorithm searches for areas of denseness and sparseness in the ecosystem, and assigns ecosystem members to clusters accordingly.

Table 1: Descriptive statistics of the four commercial platform ecosystems studied

Characteristic	Google Apps	Google Chrome	Microsoft Office365	Internet Explorer
Platform type	Productivity suite	Web browser	Productivity suite	Web browser
Entry barriers	Technical validation	Minor first-time developer fee	Compatibility and complementarity value validation	Technical validation
Partnership or certification model	Yes	No	Yes	No
App Developers with partner or certification status	73	-	278	-
Percentage of app developers with partner or certification status	7.36%	-	50.50%	-
Number of relationships per app developers	1.26	1.03	1.43	1.07
App developers with relationships	170	49	164	38
Percentage of app developers with relationships	17.14%	3.18%	29.82%	7.35%
Number of app developers	992	1539	550	517
Network density	0.0025	0.0001	0.0050	0.0004
Number of applications per app developer	1.36	1.34	2.18	1.64
App developers with one application	829	1355	368	410
Percentage of app developers with one application	83.18%	87.58%	66.85%	78.57%

The next five measures are network structure metrics. The measures indicate that there is substantial variety in the network structures of the four ecosystems studied. The first measure – *number of app developers* – expresses the size of the 'cloud' of app developers that develop one or more applications for the platform (Burkard et al., 2012). The *number of relationships per app developer* is computed as a direct count of the number of interfirm relationships that each app developer initiated, averaged across the ecosystem. Here, each app developer is assumed to engage in at least one interfirm relationship, that is, they are all connected to the platform owner. The remaining network structure metrics provide further insight into the extent to which app developers collaborate. *App developers with relationships* and *percentage of app developers with relationships* respectively provide a respectively absolute and relative measure of the rate at which app developers participate in interfirm relationships other than the one with the platform owner. The percentage of app developers that initiated interfirm relationships with at least one other app developer in the same ecosystem ranges from just over *3* percent in the Google Chrome ecosystem to almost *30* percent for Microsoft Office365.

Network density reflects the ratio of the number of interfirm relationships that are present in an ecosystem compared to the number of relationships that could theoretically be initiated (Scott, 2000). Interfirm networks with high network density have been associated with a greater degree of innovation (Schilling and Phelps, 2007) and specialization (Iyer et al., 2006). Network density is computed as

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

where E denotes the number of interfirm relationships in the ecosystem and V the number of app developers. Network density takes values between zero and one[9].

The last three measures – *average number of applications per app developer*, *number of app developers with one application*, and *percentage of developers with one application* – capture the development activity of app developers

[9] In an empty graph, no interfirm relationships are initiated, whereas in a fully connected graph all possible interfirm relationships are present (i.e., all ecosystem members are directly connected to one another).

within the ecosystem. The statistics illustrate differences among the ecosystems. For instance, the average number of applications per app developer ranges from *1.34* to *2.16*.

The computation of descriptive statistics as part of the within-case analysis was followed by a comparative cross-case analysis (Eisenhardt, 1989, 1991). We started our comparative analysis by contrasting the ecosystems of Google and Microsoft (i.e., Google Apps to Microsoft Office365 and Google Chrome to Internet Explorer). The goal here was to disentangle the way in which the network structure of ecosystems that are governed with different entry barriers and partnership and certification programs differ. Then, we compared the ecosystems that exist around different types of software platforms (i.e., enterprise productivity suites and web browsers), such that we contrasted Google Apps to Google Chrome and Microsoft Office365 to Internet Explorer. This allowed us to study the extent to which ecosystems from the same industry or domain have a similar network structure. Finally, observations from this systematic comparison exercise were contrasted to extant literature and a set of propositions was formulated.

4. Ecosystem Descriptions

In this section we present a detailed description of the four ecosystems studied. First, we describe the Google ecosystems, followed by an elaboration upon their Microsoft counterparts.

4.1. Google Apps

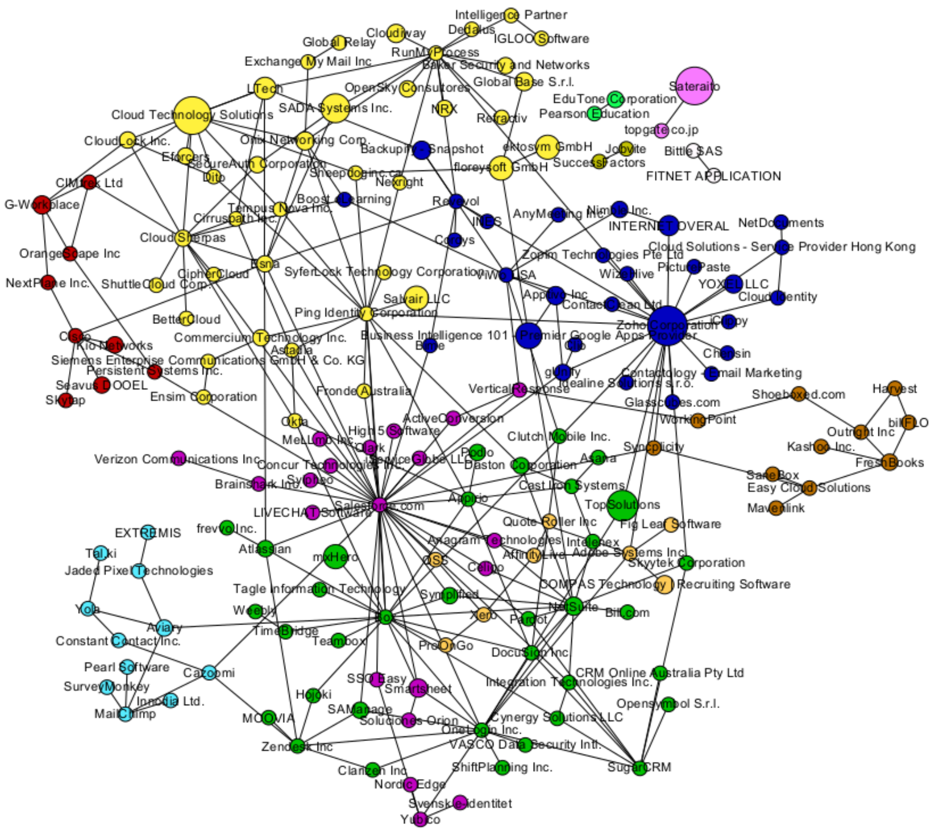

Figure 2: Cluster visualization of the Google Apps ecosystem. The node sizes are representative for the number of applications in the app store. Nodes with corresponding shades belong to the same cluster as assigned by the modularity algorithm.

The Google Apps Marketplace was introduced in 2010 as a central outlet for all third party applications for Google Apps. After an application is submitted to the Google Apps Marketplace, it is subjected to a technical validation by Google. The requirements for the technical validation are documented in the Google developer portal. At the time of data collection, *993* developers (992 app developers and Google) listed *1354* applications in the Google Apps Marketplace. Thirteen applications were developed by Google, using the aliases *'Google Inc.'* and *'Google Labs'*. The vast majority of applications was developed by companies (*96* percent), whereas the remainder is developed by individuals (*3* percent) and open source developer communities (*1* percent). From all app developers, merely 73 (7.36 percent) participate in the Google Enterprise Partner Program or one of its ancillary certification schemes. The number of applications developed per app developer ranges from one to fifteen, a complete distribution is shown in Figure 6.

Members of the ecosystem are connected by *1248* interfirm relationships, corresponding to an average of *1.26* relationships per app developer. The ecosystem has a network density of *0.25* percent, portraying a sparse degree of interconnectivity. *170* app developers participate in interfirm relationships with at least one other Google Apps app developer, the topology of the network is visualized in Figure 2.

There are four dyads that consist of two app developers that collaborate with each other, but not with other members of the ecosystem. The dyads consist of companies from the same country. *Sateraito* and *topgate.co.jp* for example are both Japanese and *Bittle SAS* and *FITNET APPLICATION* are headquartered in France. The *Salesforce.com* cluster has a hub-and-spoke network structure, presumably due to the platform efforts of *Salesforce* itself. Similarly, app developers from different geographic locations are grouped around *Zoho Corporation*, making the clusters of *Salesforce.com* and *Zoho Corporation* examples of technology clusters that are composed of members that do not necessarily collaborate because they are geographically close. Technological proximity also seems to be the binding factor in such clusters as the one with *Cloud Technology Solutions*. The majority of app developers consists of cloud service providers, including *Cloudaway*, *Cloud Sherpas*, *BetterCloud*, *CloudLock Inc.*, *Cipher Cloud*, and *ShuttleCloud Corp*. Perhaps the most appealing example of the importance of technological complementarity in larger clusters can be drawn from the *OrangeScape* cluster (visualized on the left side of Figure 2). It constitutes app developers headquartered in six different countries: the Netherlands (*G-Workplace*), United Kingdom (*CIMtrek*), United States (*Cisco*, *NextPlane*, *OrangeScape*, *Skytap*), Mexico (*Kio Networks*), India (*Persistent Systems*), and Sweden (*Seavus DOOEL*). Also noteworthy is that some of the app developers that listed the largest number of applications, such as *SaaSt* (eleven applications) and *myERP* (nine applications), did not initiate any interfirm relationships with other app developers in the Google Apps ecosystem.

4.2. Google Chrome

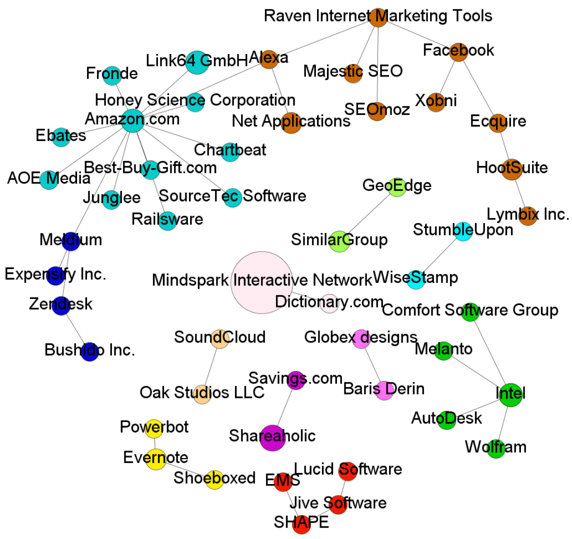

Figure 3: Cluster visualization of the Google Chrome ecosystem. The node sizes are representative for the number of applications in the app store. Nodes with corresponding shades belong to the same cluster as assigned by the modularity algorithm.

Google Chrome is the largest ecosystem investigated in this study. In May 2013, the Chrome Web Store contained *2057* applications that were developed by *1540* developers (1539 app developers and Google). Google itself is the largest distributor of extensions for the Chrome web browser as it lists *52* applications in the app store, a complete distribution is included in Figure 6. Noteworthy is the large number of individual developers (*41* percent) in the ecosystem. Companies (*32* percent), and developer communities (*18* percent) are also part of the ecosystem. Apart from a minor first-time publisher fee of $10, Google does not impose any technical or functional validation to prospective app developers for including applications in the Google Chrome Web Store.

There is little interconnectivity among app developers in the Google Chrome ecosystem. The total number of initiated interfirm relationships is equal to *1586*. Merely *49* app developers (*3.18* percent of the total population of ecosystem members) initiated at least one interfirm relationship with another Google Chrome app developer. Despite the lack of overall interconnectivity, there appears to be some degree of cohesion among ecosystem members that do partake in interfirm relationships. The interacting app developers in the ecosystem are visualized in Figure 3.

The app developers shown in Figure 3 are connected by *49* interfirm relationships. Most relationships appear to be based on the complementarity of applications offered or span across a wider scope than the software platform alone. Illustrative are the interrelationships among the app developers in the cluster depicted in the top left of the figure, which appears to harbor a number of app developers with a common specializations in social media, search engine optimization, and marketing (e.g., *Alexa*, *Raven Internet Marketing Tools*, *Majestic SEO*, and *SEOmoz*).

4.3. Microsoft Office365

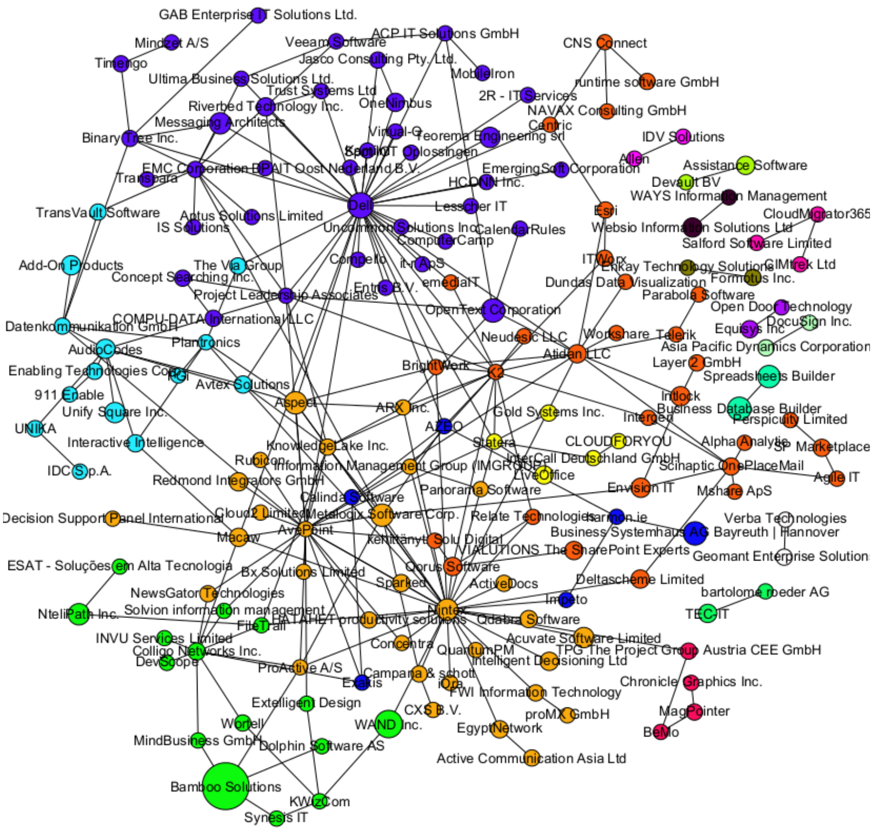

Figure 4: Cluster visualization of the Microsoft office365 ecosystem. The node sizes are representative for the number of applications in the app store. Nodes with corresponding shades belong to the same cluster as assigned by the modularity algorithm.

The Office365 Marketplace contained *1204* applications developed by *550* app developers, of which 278 (50.50 percent) are participating in the Microsoft Certified Partner Network. For an application to be included in the Office365 Marketplace, it has to be subjected to technical compatibility and complementary value requirements where Microsoft reserves the right to refuse inclusion of applications that do not provide direct added-value to the software platform. On average, each app developer listed *2.18* applications with a standard deviation of *1.65* applications. The ecosystem is predominantly populated by companies (*98* percent) and few communities or individuals (*2* percent). Noteworthy is that Microsoft itself is not involved in the development of applications for Office365, as opposed to the Google ecosystems discussed previously. The most prominent app developer in the ecosystem is *Net2xs* that listed *39* applications, followed by *Bamboo Solutions*, and *Orlando's VBA and Excel Site* that each produced *32* applications. Meanwhile, *67* percent of app developers listed just one application. A complete distribution of these descriptives is included in Figure 6.

The ecosystem is connected by *787* interfirm relationships. On average, every app developer initiated *1.43* relationships with a standard deviation of *11.74* relationships. When discarding the interfirm relationships initiated with the platform owner, app developers are connected through *0.43* relationships per app developer (standard deviation of *1.37* interfirm relationships). *Dell* (eleven applications) with *37* relationships and *Nintex* (seven applications) with *32* relationships are the most interconnected members of the ecosystem. With a network density of *0.50* percent, the Office365 ecosystem is more densely interwoven than the previously discussed ecosystems. Yet, *70.18* percent of app developers are solely connected to Microsoft and found in the periphery of the ecosystem.

Figure 4 visualizes the interfirm relationships among *164* (*29.82* percent) app developers. Noteworthy is that many of the app developers that listed the most applications are present in the network visualization. From the thirty most productive app developers, thirteen are present in Figure 4, meaning that they initiated at least one interfirm relationship with another Microsoft Office365 app developer. Closer inspection reveals that absent app developers 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 4. Interfirm relationships seem to pertain technological partnerships, as most clusters seem to lack a geographical focus. As observed previously, *Dell* is among the most well embedded app developers in the ecosystem, this is partly due to its acquisition of *Quest Software*[10].

4.4. Internet Explorer

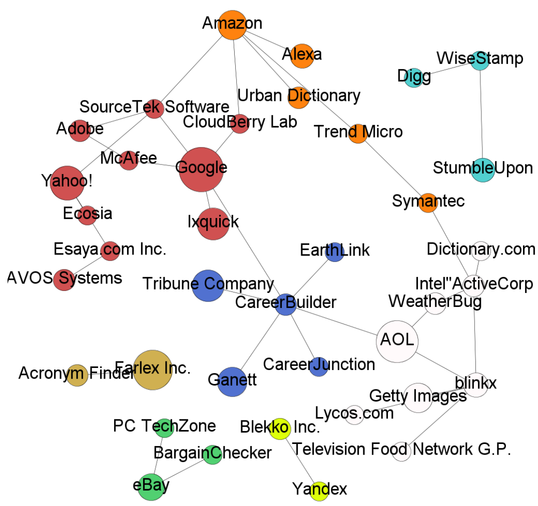

Figure 5: Cluster visualization of the Internet Explorer ecosystem. The node sizes are representative for the number of applications in the app store. Nodes with corresponding shades belong to the same cluster as assigned by the modularity algorithm

The Internet Explorer ecosystem is the smallest of the ecosystems studied. The Internet Explorer Gallery contained *853* applications developed by *518* developers (517 app developers and Microsoft). To sell a web browser extension through the Internet Explorer Gallery, the application has to pass a technical validation by the Internet Explorer

[10]http://www.dell.com/learn/us/en/uscorp1/secure/2012-09-28-dell-acquisition-quest-software

developer team. A large quantity of the applications listed in the app store is courtesy of *Brand Thunder LLC* that has developed *80* web browser extensions for sports clubs and well-known brands. Contrary to its strategy for the Microsoft Office365 ecosystem, Microsoft itself does list web browser extensions. At time of measurement, Microsoft released *22* applications. As shown in Figure 6, merely fifteen app developers develop more than five applications for Internet Explorer, reflecting limited commitment to the software platform. Ecosystem members are predominantly companies (*62* percent), followed by individuals (*11* percent), and open source communities (*9* percent).

The members of the Internet Explorer ecosystem are connected through *554* interfirm relationships, an average of *1.07* relationships per app developer. *CareerBuilder* is the most interconnected app developer with seven interfirm relationships, followed by *blinkx*, *Google*, and *Amazon* with six relationships each. Despite the sparse degree of connectivity among the members of the ecosystem, its network density is remarkably higher when compared to its equivalent Google Chrome.

Figure 5 visualizes the interfirm relationships initiated among *38* app developers. Similar to the occurrences of interfirm relationships in the Google Chrome ecosystem, the clusters in the Internet Explorer ecosystem appear to mainly present collaboration among app developers with similar, or at least related, applications. Examples include the collaborations between anti-virus software developers *Symantec* and *Trend Micro*, recruiting portals *Career Junction* and *CareerBuilder* and search engine providers *Google* and *lxquick*. Also noteworthy is the interconnectivity among the bigger clusters in the ecosystem, which are all mutually connected through linchpins.

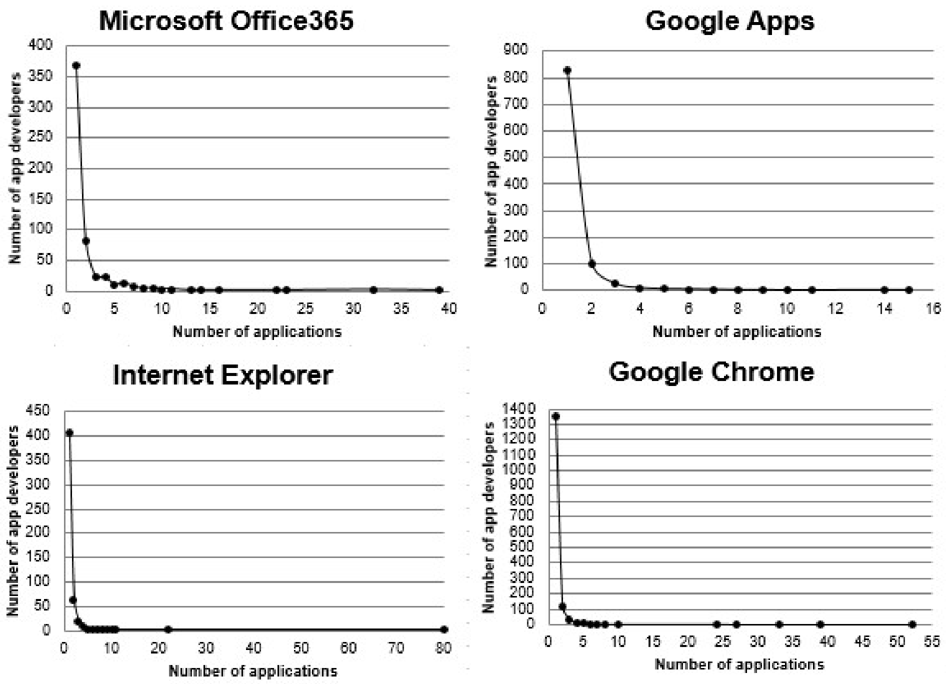

Figure 6: Distribution of app developers per platform ecosystems based on the number of applications developed

5. A Comparison of Commercial Platform Ecosystems

In the previous section we described each commercial platform ecosystem separately. In the current section, we draw comparisons between pairs of ecosystems. Our single-case analyses show that the ecosystems share several similarities. First of all, the studied ecosystems have a similar network structure. The ecosystems are highly centralized hub-and-spoke networks, and generally sparsely connected. Second, the majority of app developers only list one application within the studied ecosystems. This, may reflect a strong focus in their respective application portfolios or relative immaturity of the ecosystems. Last, all platform owners at least enforce some degree of entry barriers to govern their commercial platform ecosystem.

In this section we present an in-depth examination of observed differences in the studied ecosystems. We start by contrasting the ecosystems of Google and Microsoft (i.e., Google Apps and Microsoft Office365, Google Chrome and Internet Explorer) to study the consequences of imposing entry barriers and creating partnership models, respectively. Thereafter, we analyze the influence of inherent differences in the domain of the underpinning software platform through a comparison of ecosystems within the portfolio of Google and Microsoft, respectively.

5.1. Entry Barriers

The traditional tension between *'open'* and *'closed'* is perhaps one of the most recurrently addressed aspects of software ecosystem governance (West, 2003; Boudreau, 2010; Jansen et al., 2012). The degree of openness of an ecosystem is to a large extent determined by the entry barriers that are enforced to the inclusion of an application into the app store (Eisenmann et al., 2009). These entry barriers may encompass technical, financial, and business requirements to be met by prospective app developers.

Gawer and Cusumano (2008) and Eisenmann et al. (2009) analyzed how increased openness influences growth and variety in the ecosystem. With their open enterprise model, Jansen et al. (2012) emphasize a need for increased openness by providing software producing organizations with a model to determine how *'open'* or *'closed'* their ecosystems are. Jansen and colleagues argue that software platform selection by app developers and customers is based on certain '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 because they are deemed to foster lock-ins and abuse intellectual property rights. This in turn may discourage app developers to partake in the ecosystem at all. Other authors provide a more nuanced perspective as to the potential benefits of openness. Boudreau (2012) uses empirical evidence from the hand-held gaming industry to posit that openness directly contributes to the variety and quantity of available applications in a platform ecosystem, yet this positive relationship will only exist until a certain threshold. After this threshold is reached, the platform owner will have provoked so much competition in the ecosystem that crowd-outs of app developers will follow. Earlier, Boudreau and Hagiu (2009) already used anecdotal evidence of cases such as Facebook to illustrate that maintaining stringent entry barriers in the early days of a platform may result in increased growth of the ecosystem at a later maturity stage.

When looking at our empirical data, the *entry barriers* that are formulated by Google are more liberal compared to those of Microsoft. Google does not impose any quality appraisal for applications to be included in the Google Chrome Web Store, apart from the payment of a minor first-time publisher fee. In contrast, each application submitted to the Internet Explorer Gallery is first subjected to a technical validation. Google Apps and Microsoft Office365 are comparable in their technical requirements for third party applications. Yet, Microsoft imposes discriminatory 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 software platform.

In line with the described differences in imposed entry barriers, we identified varying *numbers of app developers* per studied ecosystem that point in the same direction. Google Chrome is the largest of the studied ecosystems with *1539* app developers, while Internet Explorer Gallery contained applications by *517* app developers. In similar vein, the Google Apps ecosystem was populated by more app developers (*992*) compared to the Microsoft Office365 ecosystem (*550* app developers). Our empirical evidence thus suggests that a relationship may exist between the level of enforced entry barriers and the number of app developers that join the ecosystem, such that imposing lower entry barriers to the app store will be positively related to the number of app developers in the ecosystem.

To summarize, in line with Gawer and Cusumano (2008), Eisenmann et al. (2009), Boudreau (2012), and others, we postulate the following relationship:

Proposition 1. *Lower entry barriers to the app store will be positively related to the number of app developers that populates a commercial platform ecosystem.*

5.2. Partnership Models

Despite the observed higher number of app developers in both Google ecosystems, their Microsoft counterparts display both greater averages for *number of applications developed per app developer* (e.g., *2.18* applications for Microsoft Office365 and *1.36* applications for Google Apps) and *number of interfirm relationships initiated per app developer* (e.g., *1.43* interfirm relationships for Microsoft Office365 and *1.26* interfirm relationships for Google Apps). These observed differences may be attributed to Microsoft's active partner development strategy. The partnership model has been proposed as a locus of control by which the platform owner enables app developers to more actively participate in the ecosystem; hereby ensuring 'coherent' productivity (Gawer and Cusumano, 2008; van Angeren et al., 2013b; den Hartigh et al., 2013). By means of their partnership model, Microsoft can foster lock-in effects, quality control through product certification (van Angeren et al., 2013b), and platform exclusivity (Boudreau and Hagiu, 2009). Ecosystems characterized by stringent governance have also been argued to display a greater degree of interconnectivity (Iyer et al., 2006; Rosenkopf and Schilling, 2007). Following this line of argumentation, one would expect that Microsoft – that is well known for its extensive Microsoft Certified Partner Network[11] – can stimulate both the number of applications developed by app developers and the network density of its platform ecosystem by means of its partnership model. Apart from the previously discussed differences in number of applications developed and number of interfirm relationships initiated between ecosystems of Google and Microsoft, this line of reasoning is reinforced by the empirical observation that the Microsoft Certified Partner Network has a wide coverage (*50.50* percent) within the Office365 ecosystem. In contrast, Google's Enterprise Partner Network merely comprises *7.36* percent of app developers in the Google Apps ecosystem.

We explore the relationship between partnership model participation and the number of applications developed, and the number of initiated relationships per app developer in the context of the Microsoft Office365 ecosystem because the number of partners and non-partners is approximately equally distributed. We divided Microsoft Office365 app developers in two cohorts based on their recorded partner status (coded by means of a dummy variable). In total, *278* app developers are certified as Microsoft partners, and *272* are non-partners. Then, we performed Mann-Whitney U tests[12] to explore whether Microsoft partners initiate more interfirm relationships or develop more applications than non-partners. The results are summarized in Table 2.

Table 2: Group comparisons by means of Mann-Whitney U tests for the Microsoft Office365 ecosystem

Dependent variable	**Group**	**N**	**Mean**	**SD**	**Min**	**Max**	**U**	**Z**
# of relationships	Partner	278	1.192	2.925	1	31	28371	-6.278***
	Non-partner	272	0.522	2.509	1	36		
# of applications	Partner	278	2.313	3.351	1	32	36576	-0.792
	Non-partner	272	2.063	3.390	1	39		
# of relationships	SharePoint	102	1.324	4.008	1	36	20937	-1.635
	Non-SharePoint	448	0.757	2.360	1	31		
# of applications	SharePoint	102	3.441	4.918	1	32	17132	-4.726***
	Non-SharePoint	448	1.904	2.835	1	39		

*** $p < 0.001$.

A Microsoft partner in the Microsoft Office365 ecosystem on average initiated *1.192* interfirm relationships with a standard deviation of *2.925*, whereas a non-partner on average had *0.522* relationships with a standard deviation equal to *2.509*. Based on these averages, there is a significant difference in initiated interfirm relationships between partners and non-partners; $U = 28371, Z = -6.278, p < 0.001$. Our empirical results suggest that Microsoft partners in the Office365 ecosystem have significantly more interfirm relationships compared to non-partners. The stringent governance of the Microsoft Office365 ecosystem by means of a partnership model thus reflects in the network density of the ecosystem, such that it increases the number of initiated interfirm relationshipsr among app developers in the ecosystem.

Different results were obtained from the second Mann-Whitney U test that we performed to investigate the relationship between partnership model participation and the number of applications developed per app developer. 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 the number of applications developed by partners and non-partners; $U = 36756, Z = -0.972, p = 0.428$. This implies

[11] http://mspartner.microsoft.com

[12] We conducted nonparametric tests because of the left-skewness of the distribution of both the number of applications developed and interfirm relationships initiated per app developer (i.e., there is a relatively large number of app developers that listed one application and initiated one interfirm relationship) as shown in Figure 6.

that Microsoft partners do not develop significantly more applications compared to non-partners, the small difference in group means may be attributed to random variation or other factors.

In an attempt to explore alternative hypotheses that may confound our observations we performed subsequent analyses. Prior work (van Angeren et al., 2014) already assessed the influence of the date of entry into the ecosystem by exploring if the age for partners and non-partners differed (i.e., the date at which their first application was included in the app store). However, no empirical support was found to support this claim. Accordingly, we can conclude that the date of entry into the ecosystem did not confound (the absence of) significant differences found earlier.

Table 3: Pair-wise comparisons (Dunn post hoc analysis after Kruskall-Wallis test) for developed applications and initiated relationships by four groups of app developers in the Microsoft Office365 ecosystem

Dependent variable	Group (I) - Group (J)	Mean difference (I-J)	Std. error difference (I-J)	Z
# of applications	NPNS - PNS	-4.532	12.526	-0.339
	NPNS - NPS	-58.676	21.101	-2.781*
	NPNS - PS	-81.801	20.113	-4.057***
	PNS - NPS	-54.423	21.101	-2.579
	PNS - PS	-77.548	20.113	-3.856**
	NPS - PS	-23.125	26.318	-0.879
# of relationships	NPNS - PNS	-73.830	12.114	-6.095***
	NPNS - NPS	-36.754	20.390	-1.083
	NPNS - PS	-80.505	19.435	-4.142***
	PNS - NPS	37.077	20.390	-1.818
	PNS - PS	-6.675	19.435	-0.343
	NPS - PS	-43.751	25.431	-1.720

* $p < 0.05$; ** $p < 0.01$; *** $p = < 0.001$.
NPNS: Non-partner and non-SharePoint app developers (N = 224)
PNS: Partner and non-SharePoint app developers (N = 224)
NPS: Non-partner and SharePoint app developers (N = 48)
PS: Partner and SharePoint app developers (N = 54)

Another factor that may explain our findings is that principal components of the Microsoft Office365 platform, such as Microsoft Office and Microsoft SharePoint, already came with an extension architecture before they were included into the Office365 platform. Consequently, existing Microsoft Office or SharePoint app developers could be overrepresented among the most connected members of the ecosystem, or among the app developers that developed the majority of applications. This in turn, could have a profound impact on the overall number of developed applications and network density of the Microsoft Office365 ecosystem. Therefore, we performed subsequent analyses related to the population of SharePoint app developers in the ecosystem, as shown in Table 2.

We identified *102* SharePoint app developers and again contrasted their recorded means for applications developed and interfirm relationships initiated to those of non-SharePoint app developers. We found a significant difference in the number of applications developed ($U = 17132, Z = -4.726, p < 0.001$) between SharePoint and non-SharePoint app developers. Conversely, the number of initiated interfirm relationships did not differ; $U = 20937, Z = -1.635, p = 0.102$.

To further explore the statistical robustness of our findings we combined our prior analyses (i.e., the Mann-Whitney U tests between Microsoft Office365 partners and non-partners and between SharePoint app developers and non-SharePoint app developers) into one statistical test. This time, we divided app developers among four groups based on their partner status and whether or not they developed applications for Microsoft SharePoint. The distribution across groups was as follows:

1. non-partners that did not develop SharePoint applications ($N = 224$);
2. Microsoft partners that did not develop SharePoint applications ($N = 224$);
3. non-partners that developed SharePoint applications ($N = 48$);
4. Microsoft partners that developed SharePoint applications ($N = 54$).

Statistical inferences by means of a Kruskall-Wallis test followed by Dunn post-hoc procedures provide confirmatory evidence for our prior observations, the results are shown in Table 3. When analyzing the number of applications developed, we found significant differences between non-partner non-SharePoint app developers ($M = 1.795, SD = 2.857$) and both non-partner SharePoint app developers ($M = 3.313, SD = 5.058$), and partner SharePoint app developers ($M = 3.556, SD = 4.836$). Moreover, we found significant differences in applications developed between partner non-SharePoint app developers ($M = 2.013, SD = 2.815$) and partner SharePoint app developers. For the number of initiated interfirm relationships we identified differences between non-Microsoft partners non-SharePoint app developers ($M = 0.3527, SD = 1.236$) and both partners non-SharePoint app developers ($M = 1.174, SD = 3.051$) and partner SharePoint app developers ($M = 1.278, SD = 2.745$).

Our empirical findings – a significant difference in the number of initiated interfirm relationships between Microsoft partners and non-partners – suggests that ecosystem governance by means of a partnership model in the Microsoft Office365 ecosystem may be positively related to its network density. Meanwhile, we found no statistical support for a difference in the number of applications developed by Microsoft partners compared to non-partners. Instead, variance in the number of applications developed by app developers could be explained by other factors, such as the presence of Microsoft SharePoint app developers in the Microsoft Office365 ecosystem.

As such, our findings both confront and confirm extant research. The observation that there may be a positive relationship between partner development and network density is in line with prior anecdotal or case-based evidence. Based on three case studies with both for-profit and non-profit platform owners, van Angeren et al. (2013b) illustrate the ways in which platform owners attempt to use their partnership model to stimulate app developers to collaborate. Examples of these strategies include organizing co-creation groups with partners from the same horizontal market, maintaining partner directories, and organizing partner conferences (Jansen et al., 2012; van Angeren et al., 2013b; Jansen and Cusumano, 2013). Our findings are also in line with research that posits a positive relationship between stringent governance (e.g., by means of architectural control) and increasing interconnectivity (Rosenkopf and Schilling, 2007). Meanwhile, our findings confront expectations that application development can be incentivized through mechanisms such as partnership models (Bosch, 2009; Popp, 2010; van Angeren et al., 2013b; den Hartigh et al., 2013). Extant literature provides two potential explanations for the absence of a relationship between partnership model participation and the number of applications developed by app developers. The first line of argumentation suggests that it is generally unlikely to move app developers beyond a certain scope and that substantive variety in the ecosystem should therefore be achieved by creating an influx of new app developers (Boudreau, 2010). Boudreau (2012) found that the number of applications developed per app developer was insensitive to competition, growth of the ecosystem, or strategic incentives in the hand-held gaming industry. The second strand of literature proposes that the efficacy of governance mechanisms such as partnership models is highly dependent on the maturity of the ecosystem. (Eisenmann et al., 2009; Jansen and Cusumano, 2013).

Revisiting our empirical evidence, we postulate the presence of a relationship between partnership model participation and the rate at which app developers initiate interfirm relationships, such that the use of a partnership model to govern a commercial platform ecosystem will be positively related to its network density. Meanwhile, we found no support to posit a relationship between partnership model participation and the rate at which app developers developed applications. Therefore:

Proposition 2. *The use of a partnership model to govern a commercial platform ecosystem will be positively related to its network density.*

Proposition 3. *There will be no relationship between the use of a partnership model to govern a commercial platform ecosystem and the number of developed applications by app developers.*

5.3. The Domain of the Underpinning Platform

Besides analyzing differences in ecosystem governance, one of the aims of our research was to compare ecosystems that exist around different types of underpinning software platforms. Therefore, we continue our inquiry by comparing the studied ecosystems based on the domain of their underpinning software platform. We compare the online productivity suites Google Apps and Microsoft Office365 and the web browsers Google Chrome and Internet Explorer. Prior studies have implicitly assumed that the type or domain of underpinning platform matters by delineating distinct types of software ecosystems. Jansen et al. (2010) for instance distinguish between ecosystems that center around a certain market, technology, platform, or firm. In similar vein, Bosch (2009) outlines operating systems, applications, and end-user programming as underpinning technologies for software ecosystems in the desktop, web, or mobile industry.

The notion that the domain of the underpinning software‘ platform has an influence on the network density of a commercial platform ecosystem can foremost be grounded in the literature that regards interfirm relationship formation in the software industry from the perspective of product complementarity (Leger and Quach, 2009; Kude et al., 2012). Literature in this arena suggests that interfirm relationships are most prevailing and successful when companies are related, e.g., companies that develop related or compatible applications, or companies that have the same set of target customers. According to Gao and Iyer (2008) this is the case because companies with related products have the most to gain from the initiation of interfirm relationships with their peers. Because they usually target the same – or at least similar – markets, related companies are most likely to exploit consumer-side synergies (i.e., serving multiple customer needs of the same market segment) and benefit from economies of scale.

Following the theory of product complementarity, one would expect that the Google Apps and Microsoft Office365 ecosystems display greater network densities compared to the Google Chrome or Internet Explorer ecosystem. The applications developed by Google Apps and Microsoft Office365 app developers are all situated in roughly the same horizontal market, targeting a fairly homogeneous set of customers that largely constitutes small to medium-sized enterprises. Indeed, the Google Apps and Microsoft Office365 ecosystems could be touted as being *'roughly similar'* in their network structure. The same holds for the Google Chrome and Internet Explorer ecosystems. Illustrative are some descriptives such as the *network densities* (respectively *0.25* percent for Google Apps and *0.50* percent for Microsoft Office365 versus *0.01* percent for Google Chrome and *0.04* percent for Internet Explorer), but the similarity of these ecosystems also resonates from the cluster visualizations that we presented in the preceding section.

Further support comes from the responses we received to our email questionnaire among Google Apps app developers. In reply to the question *"How likely is the initiation of new interfirm relationships between your company and other Google Apps app developers in the near future?"*, two respondents indicated that:

> *"Depending on customer needs and feedback we might try to create business or technical synergies with other products."* (CEO)

> *"We occasionally seek to initiate new interfirm relationships with Google Apps app developers whose applications are complementary to our own offerings. We assess the quality of their applications in accordance with our own quality criteria in before we decide to establish an interfirm relationship. These initiatives are often driven by requests of our customers."* (CEO)

Apparently, the initiation of new interfirm relationships can indeed be driven by specific needs of customers. Both respondents explicitly portray a customer request as a trigger for what could be thought of as ad-hoc strategic planning of new interfirm relationships. Meanwhile, also technological compatibility is deemed important by both respondents.

Further, it became evident from the responses that many app developers in the Google Apps ecosystem do actively seek to initiate new interfirm relationships. Numerous app developers indicated to receive new requests to collaborate on a weekly basis. One of such quotes is the following:

> *"We receive several offers to participate in partnerships with other Google Apps app developers on a weekly basis. We have created a business development team that is responsible for selecting and developing new partnerships. The team carefully assesses each request based on such criteria as technological complementarity and business value."* (Head of marketing)

Our empirical evidence suggests that customer demand for similar applications may be related to the rate at which app developers initiate interfirm relationships, such that strong demand for related products in a commercial platform ecosystem will be positively related to its network density. In commercial platform ecosystems characterized by demand for similar applications, both app developers and customers benefit most from customer-side synergies or network externalities. Following this line of argumentation, it becomes logical that Iyer et al. (2006) recorded more alliance formation activity in the SAP ecosystem compared to the IBM ecosystem. The SAP ecosystem harbors app developers that cater to similar product demands (i.e., enterprise software products). Our empirical data display a similar tendency, in that Google Apps and Microsoft Office365 ecosystems display far greater interfirm relationship activity compared to Google Chrome and Internet Explorer. Hence:

Proposition 4. *Strong customer demand for related applications in a commercial platform ecosystem will be positively related to its network density.*

6. Discussion

A widely adopted strategy for many commercial software platform owners is the cultivation of ecosystems. Platform owners attract third parties to generate a large number and rich variety of software applications. In this article we studied the characteristics of app developer relationships and the way in which enforcing entry barriers to the app store, partnership models, and the domain of the underpinning software platform may affect the initiation of these interfirm relationships. The remainder of this section further highlights the theoretical and practical contributions of our study and addresses potential limitations.

6.1. Theoretical and Practical Implications

In this article, we have explicitly addressed the call by Manikas and Hansen (2013b) to increase the number of empirical studies of commercial software ecosystems. Our study is one among a limited number of studies (Iyer et al., 2006; van Angeren et al., 2013a, 2014) that has visualized and investigated the network structure of commercial software ecosystems. As such, our research approach and case studies of four ecosystems directly contribute to broadening the body of knowledge in a, to date, novel research domain. Our study has shown that there is substantial variety in the network structure of commercial platform ecosystems. Although the overall network density of commercial platform ecosystems was found to be low, we illustrated that app developers actively collaborate and co-create through interfirm relationships such as technological partnerships and mutual product certification. We also found that the entry barriers to the app store, partnership models, and the domain of the underpinning software platform respectively affect the number of app developers in, and network density of, commercial platform ecosystems.

There are several implications of these findings. To begin, we posit that lowering entry barriers to the app store and the use of a partnership model both affect the network structure of a commercial platform ecosystem. We contribute to the growing number of studies producing frameworks that outline a set of mechanisms by which a platform owner can steer its ecosystem into a favorable direction (Gawer and Cusumano, 2008; Jansen et al., 2012; Jansen and Cusumano, 2013; Rickmann et al., 2014). So far, this stream of research has largely used anecdotal or incidental case-based evidence to speculate about the efficacy of varying entry barriers and creating partnership models. Gawer and Cusumano (2008) structured governance along 'levers' of managerial decisions that encompassed defining the scope of the focal firm, technological management of the underpinning platform, managing relationships with app developers, and internal organization. The authors argued that a combination of all these strategies would lead to optimal results. Jansen and Cusumano (2013) took a more performance-oriented perspective in structuring governance mechanisms along the constituents of ecosystem health that they may affect. Rickmann et al. (2014) posit that ecosystem governance should provide enablers and instruments to aid app developers in achieving their goals. Our study highlights the importance of research that investigates the efficacy of software ecosystem governance mechanisms. This necessity is for instance illustrated by our inability to demonstrate the presence of a relationship between the use of a partnership model and the number of applications developed by app developers. A relationship that to date has been recurrently anticipated in extant literature (Bosch, 2009; Popp, 2010; van Angeren et al., 2013b; den Hartigh et al., 2013).

Our findings also have distinct implications for the future development of operational metrics for software ecosystem health measurement. In line with studies by den Hartigh et al. (2013) and Monteith et al. (2014), we conclude that the operationalization of such metrics is troublesome and highly situational. Our empirical results emphasize the need to develop specific ecosystem health frameworks for different types of software ecosystems, such as the currently observable distinction between open source and commercial ecosystem health (Manikas and Hansen, 2013a; Jansen, 2014). By building on the theory of product complementarity (Gao and Iyer, 2008; Leger and Quach, 2009; Kude et al., 2012), we posited that an ecosystem populated by app developers that develop related applications for a fairly homogeneous set of target customers (e.g., Google Apps and Microsoft Office365) are likely to have a greater network density because of the greater gain from customer-side synergies and network externalities. This finding – that the domain of the underpinning software platform is related to the network density of a commercial platform ecosystem – suggests that some metrics might be more relevant or appropriate to some ecosystems than to others.

This finding also portrays part of the managerial implications of our research. Our study underscores the influence that the distinct characteristics of a commercial platform ecosystem can have on the efficacy of the ecosystem governance strategy formulated by a platform owner, or the partner strategy by app developers. From the perspective of the platform owner, our findings imply that a governance strategy that is effective in one software ecosystem may prove unsuccessful in another. At the same time, our study illustrates the efficacy of two specific governance mechanisms, being enforcing entry barriers to the app store and using partnership models. To summarize, the following findings provide managerial insights for platform owners:

- the efficacy of enforcing entry barriers and creating partnership models differs across types of software ecosystems;
- low entry barriers to the app store can result in growth or increased variety in app developers;
- partner development can foster the initiation of interfirm relationships in a commercial platform ecosystem;
- software ecosystem health metrics need to be individually formulated for each ecosystem.

To the benefit of app developers, our study advocates that it is important to be aware in what sort of ecosystem one operates. Actively forging interfirm relationships to strengthen one's position might only prove to be a more viable or necessary entry strategy for an app developer if the software ecosystem is characterized by demand for related products.

6.2. Limitations

Similar to any exploratory case study, our study has some limitations. In the absence of established data collection procedures to conduct empirical research on commercial software ecosystems, choices had to be made to provide a holistic reconstruction of the network structure of four commercial platform ecosystems. We limited our scope to those third parties that developed one or more applications for the underpinning software platform under study and the interfirm relationships among them. We treated interfirm relationships as symmetric and binary ties. Further, we relied on data collected directly from websites of app developers that are inherently incomplete and ambiguous. For instance, some app developers indicated to partake in interfirm relationships, yet they did not provide a list of their actual partners on their company websites.

Because prior usage of a similar data collection method has been scarce in extant literature, we evaluated the accuracy and completeness of our method with a subset of Google Apps app developers. Based on the results of an email questionnaire, we preliminarily concluded that our data collection procedure is reliable. However, future research should demonstrate if our findings are robust to further enhancements of the data collection procedures. In addition, other ecosystems should be investigated using the same research approach. More granular analyses could be performed by delineating between different types of interfirm relationships (i.e., technological partnerships, partnership model participation, mutual product certification, and others) or by including directional relationships (e.g., Salesforce.com considers Box as a partner, but Box does not list Salesforce.com as a partner) into the study.

The findings presented in this article were derived from cross-sectional network data, in that it only represents a snapshot of each commercial platform ecosystem constructed at a given point in time. Therefore, any conclusion drawn from this data is inherently time-variant. Imputations about the evolution of the ecosystem and how entry barriers to the app store, partnership models, or the domain of the underpinning platform may influence the number of app developers in, or the network density of, the ecosystem over time were based on either observed differences in the ecosystems or extant literature. Future research should combine multiple snapshots of the same ecosystems to further investigate the influence of the factors that we identified.

7. Conclusion and Directions for Future Research

In this article we presented the results of an inductive multiple case study of commercial platform ecosystems of Google (Google Apps and Google Chrome) and Microsoft (Microsoft Office365 and Internet Explorer). We used data collected from the app stores, directly from the websites of app developers, and CrunchBase to reconstruct and visualize the network structure of these ecosystems in order to study interfirm relationships among app developers.

We found substantial variety in network structure among the studied ecosystems. In particular, we investigated how entry barriers to the app store, partnership models, and the domain of the underpinning software platform respectively impact the number of app developers and applications developed in, and network density of, a commercial platform ecosystem. Consistent with other studies (e.g., Gawer and Cusumano, 2008; Eisenmann et al., 2009; Boudreau, 2012; Jansen et al., 2012), we our empirical evidence suggests that lowering entry barriers to the app store will be positively related to the number of app developers in the ecosystem. The use of a partnership model to govern a software ecosystem was found to be positively related to the network density of a commercial platform ecosystem, yet the average number of applications per app developers remained unaffected. We built on the theory of product complementarity (Gao and Iyer, 2008; Leger and Quach, 2009; Kude et al., 2012) – which suggests that interfirm relationships are more prevalent when companies develop complementary applications for similar market segments – combined with empirical evidence to posit that the domain of the underpinning software platform of a commercial software ecosystem is related to its network density. Ecosystems characterized by stronger demand for related applications (Google Apps and Microsoft Office365) were found to display a greater network density when compared to ecosystems without such demand.

The findings presented in this article contribute to advancing the understanding of the implications of software ecosystem governance. Our study is among the first to empirically assess the efficacy of two software ecosystem governance mechanisms (i.e., enforcing entry barriers to the app store and creating partnership models) in relation to properties of commercial platform ecosystems. Based on inductive empirical evidence, we posit that the use of entry barriers and partnership models is directly related to the number of app developers in, and network density of, a commercial software ecosystem. To the benefit of platform owners, our research shows that the efficacy of entry barriers and partnership models is dependent on the domain of the software platform that underpins the ecosystem. Consequently, our findings indicate that ecosystem governance mechanisms should be selected for each specific ecosystem, rather than formulating a unified ecosystem governance strategy for all software platforms in the portfolio of a platform owner.

The presented study leaves many avenues for future research. Foremost, longitudinal case studies should be performed to test the propositions that we developed throughout this article. Our data collection and analysis procedures may serve as a starting point or even as a blueprint for these studies. Novel ways for enhancing our data collection and analyses could be sought. Replication studies could expand the scope of this study by explicitly accounting for 'multi-homing' (Landsman and Stremersch, 2011); a phenomenon that is prevalent in almost any segment in the software industry. Since multi-homing app developers are less likely to intensively commit themselves to one ecosystem, a particularly interesting question would be to explore the impact of multi-homing on the intensity of interfirm relationship initiation in commercial platform ecosystems. Further, different data collection techniques, such as more in-depth interviews or surveys among app developers could be included to improve triangulation strategies. From a managerial point of view, further automation of the data collection process could contribute to the potential implementation of ecosystem analysis in partner development. This would allow platform owners to monitor the position of individual app developers in the ecosystem to act accordingly. The inclusion of other types of ecosystem members (e.g., service providers, resellers, and others) into the scope of future studies could contribute towards providing a more holistic perspective of commercial platform ecosystems. Such a perspective could also aid platform owners or app developers to use 'structural hole' analysis (Ahuja, 2000) as means to identify potential niches, for example when multiple app developers are not yet surrounded by any service provider or vice versa.

This article can also be read as a call for action to advance the research agenda on commercial software ecosystems. Existing frameworks for software ecosystem governance combined with the findings of this article could be used as a starting point to perform more in-depth evaluations of the efficacy of governance mechanisms. Software ecosystem governance frameworks could be adapted with a situational perspective of the mechanisms that are most suitable in a certain type or maturity stage of a commercial software ecosystem. As such, future research could better define the strategic toolbox of ecosystem governance and orchestration for (prospective) platform owners.

References

Ahuja, G., 2000. Collaboration Networks, Structural Holes, and Innovation: A Longitudinal Study. Administrative Science Quarterly 45 (3), 425–455.

Basole, R. C., 2009. Visualization of interfirm relations in a converging mobile ecosystem. Journal of Information Technology 24 (2), 144–159.

Basole, R. C., Karla, J., 2011. On the evolution of mboile platform ecosystem structure and strategy. Business and Information Systems Engineering 3 (5), 312–322.

Basole, R. C., Park, H., Barnett, B. C., 2014. Coopetition and convergence in the ICT ecosystem. Telecommunications Policy 36 (7), 537–552.

Bastian, M., Heymann, S., Jacomy, M., 2009. Gephi: An Open Source Software for Exploring and Mapping Networks. Proceedings of the Third AAAI Conference on Weblogs and Social Media, 361–362.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., Lefebre, E., 2008. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 5 (10).

Borgatti, S. P., Everett, M. G., Freeman, L. C., 2002. Ucinet for Windows: Software for Social Network Analysis. Analytic Technologies, Harvard, MA, USA.

Bosch, J., 2009. From software product lines to software ecosystems. Proceedings of the Thirteenth International Software Product Line Conference, 1–10.

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

Boudreau, K. J., 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.

Boudreau, K. J., Hagiu, A., 2009. Platform rules: Multi-sided platforms as regulators. In: Gawer, A. (Ed.), Platforms, Markets and Innovation. Edward Elgar Publishing, Cheltenham, UK, pp. 163–191.

Burkard, C., Widjaja, T., Buxmann, P., 2012. Software ecosystems. Business and Information Systems Engineering 4 (1), 41–44.

Ceccagnoli, M., Forman, C., Huang, F., Wu, D. J., 2011. Cocreation of value in a platform ecosystem: The case of enterprise software. MIS Quarterly 36 (1), 263–290.

Chellappa, R. K., Saraf, N., 2010. Alliances, rivalry, and firm performance in enterprise systems software markets: A social network approach. Information Systems Research 21 (4), 849–871.

Chesbrough, H., 2003. Open Innovation: The New Imparative for Creating and Profiting from Technology. Harvard Business School Press, Boston, MA, USA.

Crowston, K., Howison, J., 2005. The social structure of free and open source software development. First Monday 10 (2).

den Hartigh, E., Visscher, W., Tol, M., Salas, A. J., 2013. Measuring the health of a business ecosystem. 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, pp. 221–246.

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

Eisenhardt, K. M., 1991. Better stories and better constructs: The case for rigor and comparative logic. Academy of Management Review 16 (3), 620–627.

Eisenmann, T., Parker, G., Van Alstyne, M. W., 2009. Opening platforms: How, when and why? In: Gawer, A. (Ed.), Platforms, Markets and Innovation. Edward Elgar Publishing, Cheltenham, UK, pp. 131–162.

Gao, L. S., Iyer, B., 2008. Partnerships Between Software Firms: Is There Value from Complementarities. Proceedings of the 41st Hawaii International Conference on System Sciences.

Gawer, A., 2009. Platform dynamics and strategies: From products to services. In: Gawer, A. (Ed.), Platforms, Markets and Innovation. Edward Elgar Publishing, Cheltenham, UK, pp. 45–76.

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

Ghazawneh, A., Henfridsson, O., 2013. Balancing platform control and external contribution in third-party development: the boundary resources model. Information Systems Journal 23 (2), 173–192.

Grewal, R., Lilien, G. L., Mallapragada, G., 2006. Location, Location, Location: How Network Embeddedness Affects Project Success in Open Source Systems. Management Science 52 (7), 1043–1056.

Hanssen, G. K., 2012. A longitudinal case study of an emerging software ecosystem: Implications for practice and theory. Journal of Systems and Software 85 (7), 1455–1466.

Iansiti, M., Levien, R., 2004a. The Keystone Advantage: What the New Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability. Harvard Business School Press, Boston, MA, USA.

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

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

Jansen, S., 2014. Measuring the health of open source software ecosystems: Beyond the scope of project health. Information and Software Technology 56 (11), 1508–1519.

Jansen, S., Brinkkemper, S., Finkelstein, A., 2010. Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems. Proceedings of the Second International Workshop on Software Ecosystems, 34–48.

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

Jansen, S., Cusumano, M. A., 2013. Defining software ecosystems: a survey of software platform and business network governance. 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, pp. 13–28.

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.

Kagdi, H., Collard, M. L., Mealetic, J. I., 2007. A survey and taxonomy of approaches for mining software repositories in the context of software evolution. Journal of Software Maintenance and Evolution: Research and Practice 19 (2), 77–131.

Katz, M. L., Shapiro, C., 1985. Network Externalities, Competition, and Compatibility. American Economic Review 75 (3), 424–440.

Kazman, R., Gagliardi, M., Wood, W., 2012. Scaling up software architecture analysis. Journal of Systems and Software 86 (7), 1511–1519.

Koch, S., Kerschbaum, M., 2014. Joining a smartphone ecosystem: application developers' motivations and decision criteria. Information and Software Technology 56 (11), 1423–1435.

Kude, T., Dibbern, J., Henzl, A., 2012. Why Do Complementors Participate? An Analysis of Partner Networks in the Enterprise Software Industry. IEEE Transactions on Engineering Management 59 (2), 250–265.

Landsman, V., Stremersch, S., 2011. Multihoming in Two-Sided Markets: An Empirical Inquiry in the Video Game Console Industry. journal of Marketing 75 (6), 39–54.

Leger, P.-M., Quach, L., 2009. Post-merger performance in the software industry: The impact of characteristics of the software product portfolio. Technovation 29 (10), 704–713.

Lungu, M., Lanza, M., Girba, T., Robbes, R., 2010. The Small Project Observatory: Visualizing software ecosystems. Science of Computer Programming 75 (4), 264–275.

Madey, G., Freeh, V., Tynan, R., 2002. The open source software development phenomenon: An analysis based on social network theory. Proceedings of the Eighth Americas Conference on Information Systems, 1806–1813.

Manikas, K., Hansen, K. M., 2013a. Reviewing the health of software ecosystems – a conceptual framework proposal. Proceedings of the Fifth International Workshop on Software Ecosystems, 33–44.

Manikas, K., Hansen, K. M., 2013b. Software ecosystems – a systematic literature review. Journal of Systems and Software 86 (5), 1294–1308.

Monteith, J. Y., McGregor, J. D., Ingram, J. E., 2014. Proposed metrics on ecosystem health. Proceedings of the International Workshop on Software-defined Ecosystems, 33–36.

Moore, J. F., 1993. Predators and Prey: A new Ecology of Competition. Harvard Business Review 73 (3), 75–83.

Orton, J. D., Weick, K. E., 1990. Loosely Coupled Systems: A Reconceptualization. Academy of Management Review 15 (2), 203–223.

Popp, K.-M., 2010. Goals of Software Vendors for Partner Ecosystems - A Practitionerś View. Proceedings of the First International Conference on Software Business, 181–186.

Rickmann, T., Wenzel, S., Fischbach, K., 2014. Software Ecosystem Orchestration: The Perspective of Complementors. Proceedings of the Twentieth Americas Conference on Information Systems.

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

Schilling, M. A., Phelps, C. C., 2007. Interfirm Collaboration in Networks: The Impact of Large-Scale Network Structure on Firm Innovation. Management Science 53 (7), 1113–1126.

Scott, J. G., 2000. Social Network Analysis: A Handbook, 2nd Edition. Sage Publications, Inc., Gateshead, UK.

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

van Angeren, J., Jansen, S., Brinkkemper, S., 2014. Exploring the Relationship between Platform Strategy and Interfirm Network Structure: An Analysis of the Office365 Ecosystem. Proceedings of the Fifth International Conference on Software Business, 1–15.

van Angeren, J., Kabbedijk, J., Popp, K.-M., Jansen, S., 2013b. 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, pp. 85–102.

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

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

Joey van Angeren is a doctoral candidate at the sub-department of Innovation, Technology Entrepreneurship, and Marketing at the School of Industrial Engineering at Eindhoven University of Technology. He received his MSc. (cum laude) in Business Informatics from Utrecht University in 2013. Among his research interests are business and software ecosystems, business models, platform strategy, and product management. So far, Joey has published papers on partnership models, software ecosystems, and software product management. His doctoral dissertation directs itself at revenue model configuration in the mobile app industry.

Carina Alves is an assistant professor at the Center of Informatics at Federal University of Pernambuco, Brazil. Currently, she is a visiting researcher at Utrecht University. She received her PhD in Computer Science from University College London in 2005. She has served as guest editor of a special issue on Software Ecosystems of the Journal Information and Software Technology (November 2014). She is co-organizing the International Workshop on Software Ecosystems since 2010. Her main research interests are software ecosystems, business process management and empirical software engineering.

Slinger Jansen is an assistant professor at the Department of Information and Computing Sciences at Utrecht University. He is one of the leading researchers in the domain of software ecosystems and co-founders of the International Conference on Software Business and International Workshop on Software Ecosystems. He is lead editor of the book “Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry” and of several others. Besides his academic endeavors he actively supports new enterprises and sits on the boards of advisors of several start-ups, one of which is ThinkEcosystems.com.

***Biography**

Joey van Angeren is a doctoral candidate at the sub-department of Innovation, Technology Entrepreneurship, and Marketing at the School of Industrial Engineering at Eindhoven University of Technology. He received his MSc. (cum laude) in Business Informatics from Utrecht University in 2013. Among his research interests are business and software ecosystems, business models, platform strategy, and product management. So far, Joey has published papers on partnership models, software ecosystems, and software product management. His doctoral dissertation directs itself at revenue model configuration in the mobile app industry.

Carina Alves is an assistant professor at the Center of Informatics at Federal University of Pernambuco, Brazil. Currently, she is a visiting researcher at Utrecht University. She received her PhD in Computer Science from University College London in 2005. She has served as guest editor of a special issue on Software Ecosystems of the Journal Information and Software Technology (November 2014). She is co-organizing the International Workshop on Software Ecosystems since 2010. Her main research interests are software ecosystems, business process management and empirical software engineering.

Slinger Jansen is an assistant professor at the Department of Information and Computing Sciences at Utrecht University. He is one of the leading researchers in the domain of software ecosystems and co-founders of the International Conference on Software Business and International Workshop on Software Ecosystems. He is lead editor of the book "Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry" and of several others. Besides his academic endeavors he actively supports new enterprises and sits on the boards of advisors of several start-ups, one of which is ThinkEcosystems.com.