Coring on Digital Platforms – Fundamentals and Examples from the Mobile Device Sector

Completed Research Paper

Benedict Bender
Chair of Business Informatics,
esp. Processes and Systems
University of Potsdam
August-Bebel-Str. 89,
14482 Potsdam, Germany
benedict.bender@wi.uni-potsdam.de

Norbert Gronau
Chair of Business Informatics,
esp. Processes and Systems
University of Potsdam
August-Bebel-Str. 89,
14482 Potsdam, Germany
norbert.gronau@wi.uni-potsdam.de

Abstract

Today's mobile devices are part of powerful business ecosystems, which usually involve digital platforms. To better understand the complex phenomenon of coring and related dynamics, this paper presents a case study comparing iMessage as part of Apple's iOS and WhatsApp. Specifically, it investigates activities regarding platform coring, as the integration of several functionalities provided by third-party applications in the platform core. The paper makes three contributions. First, a systematization of coring activities is developed. Coring modes are differentiated by the amount of coring and application maintenance. Second, the case study revealed that the phenomenon of platform coring is present on digital platforms for mobile devices. Third, the fundamentals of coring are discussed as a first step towards theoretical development. Even though coring constitutes a potential threat for third-party developers regarding their functional differentiation, an idea of what a beneficial partnership incorporating coring activities could look like is developed here.

Keywords: Coring, Digital Platforms, Digital Marketplaces, Mobile Software Ecosystems

Introduction

Mobile devices such as smartphones and tablets are part of our daily business as well as private lives. Recently, mobile devices have been integrated into powerful business ecosystems, which typically involve digital platforms. These digital platforms and related marketplaces allow customers to download additional software packages for their devices to enhance the basic functionalities provided by the operating system.

In a simplified manner, software functionality is provided by the operating system of the platform owner (e.g. Apple and iOS, Google and Android) and the additional applications developed by independent software vendors (ISVs) that are distributed through their marketplaces. Platform owners provide various resources that allow third parties to develop applications for their platform. These resources typically include software development kits (SDKs) and application programming interfaces (APIs), which enable ISVs to develop additional software packages, often called 'apps.' By adjusting the aforementioned boundary resources, the platform owner retains fine-grained control over the possibilities of the third-party developer.

Innovation is of great importance on digital platforms and often constitutes a critical success factor. This is valid for the platform core of the platform owners, as well as the applications developed by the ISVs. Different modes of innovation are present on digital platforms. In contrast to the typical organization-centric considerations of innovation, the focus here lies on the interplay of different entities involved in these complex business ecosystems. These entities include, among others, platform owners, customers and independent software developers. This paper focuses on innovation activities between the platform core provided by the platform owner and the applications provided by the third-party developer. In this context, the concept of platform coring refers to the integration of functions in the platform core. This paper evaluates whether those integrated functionalities have previously been provided by corresponding applications from third-parties.

To better understand the platform core's progress, this paper examines the coring phenomenon on Apple's iOS, which serves as a prominent example of a digital platform for mobile devices. Specifically, activities regarding platform coring as the integration of several functionalities provided by third-party applications in the platform core are examined. The comparison is conducted with WhatsApp as a commonly used messaging application. The analysis uses the release notes as well as complementary sources to analyze the functional evolution of Apple's mobile operating system iOS, referred to as 'platform core', and the third-party application WhatsApp Messenger. The analysis thereby focuses on the text messaging and (video) call functionality, which are provided by both software components.

The following section summarizes the related literature as well as the theoretical foundation for the analysis. The third section then presents a systematization of coring activities, while section four explains the case study. Thereafter, the fundamentals of coring are presented and discussed for the example case. Finally, limitations as well as future research ideas are highlighted in the last section, followed by some concluding remarks.

Related literature

Digital Platforms and Business Ecosystems

Nowadays, various digital devices are equipped with access to a digital platform by default. Typical examples involve smartphones or tablets, which are embedded in the provider's digital platform, allowing them to download additional software components to enhance the basic functionality provided by the platform core (Tiwana et al. 2010). Usually, the platform core is integrated in the device's firmware and therefore is an integral part of the digital product. The basic functionality provided by the platform core is expected to be relevant for the mass market. In the case of smartphones, basic functionalities involve examples like the possibility of making calls, writing text messages, managing contacts and browsing the web. These are provided by all major smartphone operating systems like Apple's iOS, Google's Android, Microsoft's Windows Phone or RIM's Blackberry OS. Commonly, the digital platform is bound to specialized hardware (e.g., Apple iPhone) or a software operating system (e.g., Android). Additional and specialized functionality – usually developed by third parties and made available through applications – can be accessed through the digital platform, often in the form of an application store (Boudreau 2012).

Digital platforms are accompanied by complex business ecosystems that involve various parties who contribute to the ecosystem and are therefore important for the viability of the system. The integration of multiple partners in the ecosystem allows it to provide more functions than a single platform owner could provide and furthermore contributes to the scaling of the platform (Eisenmann 2011). Three parties are of special relevance for this contribution: the platform owner, the partners and the users. The *platform owner* represents the company issuing and running the platform. Using the example of smartphones, this is typically the provider of the device operating system. In some cases, the platform owner and the device manufacturer fall under the same entity (e.g. Apple with the iPhone running iOS and the AppStore), while in other cases they are represented by different entities (e.g. Android, which runs on devices of multiple manufacturers, and the PlayStore) (Kenney and Pon 2011). Providing a digital platform in the context of mobile devices involves at least (a) an infrastructure to access the software packages and (b) a standardized access for partners to develop additional software packages (boundary resources) for the platform or to contribute non-software content (Gawer and Cusumano 2008). The *partners* contribute to the ecosystem by providing complements to the digital platform. One important type of the latter are software components and the corresponding partners that are often referred to as independent software vendors (Antero et al. 2013; Ceccagnoli et al. 2012). The input of the partners helps to increase the diversity in the offering and therefore enhances customer value of the digital platform (Hyrynsalmi et al. 2016; Ghazawneh and Henfridsson 2015, 2013). The group of *users* utilizes the digital platform to access the content provided by the partners and the platform owner. When a user buys a paid application on the platform, the platform owner retains a portion of the revenue and forwards the rest to the corresponding developer.

Regarding innovation as the introduction of new functionalities, the possibilities of the different parties involved in digital platforms vary. The platform owner has the greatest flexibility and the position to control other innovation efforts on the platform (Eaton et al. 2015). The partners can innovate within the boundaries set by the platform owner using boundary resources (Ghazawneh and Henfridsson 2013). Finally, users can provide valuable input to the platform owner and third-party developers and thereby guide the innovation direction (Wong 2016; Koch and Bierbamer 2016; Kankanhalli et al. 2015). Innovation as a practice that enhances the functionality and value for platform users is a critical success factor for digital platforms (Wong et al. 2016). Innovation is often seen as a differentiating factor between competing ecosystems that allows to gain competitive advantage (Kankanhalli et al. 2015). Furthermore, innovation is a necessary condition for partners (external innovators) to participate in an ecosystem (Mohagheghzadeh and Svahn 2016; Gawer and Cusumano 2014). Their motivation to innovate on a (specific) platform is supported by the positive environment that platform leaders provide. This involves the possibility of entry into complementary markets through the use of existing structures as well as a distribution channel that provides demand for their products (Gawer and Cusumano 2014). Maintaining an optimal degree of control, while providing enough flexibility to foster innovation and ensuring the partners' motivation to participate poses a crucial challenge to the platform owner (Mohagheghzadeh and Svahn 2016; Goldbach and Benlian 2015; Ghazawneh and Henfridsson 2013). Platform owners control the opportunities for partners to innovate by adjusting the boundary resources of their platform. Various specifications of boundary resources as well as their operationalization exist (Benlian et al. 2015; Eaton et al. 2015; Ghazawneh and Henfridsson 2013).

Motivation of third-party developers is critical for the viability of the platform and the corresponding ecosystem. Furthermore, several governance aspects (e.g., compensation mechanisms) are of great importance for partners' motivation as well (Kankanhalli 2015; Oh et al. 2015; Hsieh and Hsieh 2013). While various streams of literature indicate a positive impact of digital platforms on innovation, the dominant position of various prominent and powerful platform leaders raises the question whether a negative impact on competition exists (Gawer and Cusumano 2014). Maintaining both collaboration and competition at the same time is crucial, yet difficult to balance for platform owners (Ghazawneh and Henfridsson 2013; Tiwana et al. 2010). Both aspects are essential to maintain progress and push the platform forward. As platforms progress, owners are often faced with opportunities to enter complementary markets where partners are already active. Even though these markets might offer valuable options to the owner, entering those markets creates disincentives for existing partners to further innovate (Gawer and Cusumano 2014). The execution of coring activities by platform owners can be interpreted as such an undertaking, depending on the extent and type of coring, which constitutes a potential threat for platform partners since their offering is likely to become obsolete. Once the platform core provides similar functionality, the additional application loses its functional advantage. These applications may retain their competitive advantage due to network effects through the established user base. Consequently, such decisions must be made very carefully to maintain a viable and competitive ecosystem.

Coring on Digital Platforms

In the context of digital platforms, the concept of coring is understood differently depending on the individual research focus. The following section briefly outlines different understandings of this concept.

Gawer and Cusumano (2008) use coring to subsume activities related to the creation of a platform and the design of the connected platform core. The platform core is understood as the most central and fundamental system element. The activity of coring in this context is understood as the setup of a platform in contrast to platform tipping, which refers to activities that aim at winning a war of competitive platforms (Gawer and Cusumano 2008). Gawer and Cusumano (2014) suppose that once a firm has successfully established a platform, the central position will allow for control but at the same time require the platform owner to conduct platform leadership to stay competitive.

Toppenberg et al. (2016) discuss coring in the context of external innovations through acquisitions. Coring acquisitions involve the business acquisition of a formerly external company as well as the activities necessary to integrate the innovative technologies provided by those companies into the platform core. The example of Cisco is used to investigate the value creation in the coring acquisition process. Besides the positive effects of innovation for customers, the importance of innovation for the satisfaction and constant engagement of partners to contribute components to the platform is highlighted (Toppenberg et al. 2016).

Saarikko (2016) challenges the previous understandings of coring by raising concerns about whether or not there is just one digital core and whether or not the activity can be attributed to a single company (Saarikko 2016). Using a case-study approach, an analysis of a business-to-business service provider that established a digital platform was conducted. The author describes platform coring as an emergent multi-party process, which involves the negotiation and agreement of value creation in a specific constitution. Similar to the notion of Gawer and Cusumano (2008), the focus here lies on the emerging process of the platform. This contrasts with the approach of conducting specific activities when the platform already exists, as taken by Toppenberg et al. (2016).

Um and Yoo (2016) analyze the evolution of digital ecosystems using the example of WordPress. In contrast to previous studies, WordPress as an open-source software has no central authority in the sense of business ecosystems. The authors found out that external APIs play a critical role with respect to functional heterogeneity in the growth of digital ecosystems. More specifically, they suggest that functional value is more important than the number of APIs. Previous studies show that complementary components can be clustered using common APIs (Um et al. 2013).

In the context of this contribution, coring is understood as the integration of functions into the platform core that have not been a part of the platform core before. The platform core is the most central and substantial element within the ecosystem. Taking mobile devices into consideration, this primarily entails the release of new software versions (updates) of the operating system with enhanced functionalities. For this study, the focus is narrowed to existing applications within the business ecosystem, especially in the application marketplace from which functions are transferred to the platform core.

Related Effects

This section briefly highlights some effects that are present in the partnerships around digital platforms. The focus is devoted to related effects that are especially relevant for coring, which means in the partnership between the platform owner and third-party developer, and that allow differentiation between different types of coring, which are described afterwards in the coring systematization section.

Competition or the process of competing according to the Merriam-Webster Dictionary is defined as "the effort of two or more parties acting independently to secure the business of a third party by offering the most favorable terms". Besides competition in general, different types are distinguished by various taxonomies. As an example, the differentiation between direct and indirect competition is named. In the context of digital platforms, competition is present, since platform owners and third-parties offer similar products for the same target group and are therefore in direct competition for customers, as the case study exemplifies. Competition is also present on the platform level (Armstrong and Wright 2007; Gawer and Cusumano 2014).

The term *coevolution* originated in biology, where the term is used to describe the reciprocal influence of at least two species' evolution (Darwin 1859). The term nowadays is used to describe similar effects in other contexts, e.g. business or technological contexts. Especially in the intersection of the two contexts, the term is also associated with the business ecosystem literature (Tiwana et al. 2010; Rai and Tang 2014). The mutual influence does not necessarily need to be coordinated or controlled. The effects might be a result of interaction in the same markets or through common customers of products or services. Coevolution is present on digital platforms in various forms (e.g. Tiwana et al. 2010). Many interactions and partnerships are present between different parties in the context of digital platforms and related business ecosystems.

The term *co-opetition* refers to the coexistence of competition and cooperation in a partnership between at least two independent parties. The concept of coopetition was introduced by Nalebuff and Brandenburger (1997). The idea of added value from a synergistic relationship between the parties to jointly realize a common goal refers to cooperation and value creation, while competition refers to value capturing. The added value and the mutually beneficial exchanges constitute important motivation factors for participation. The ambivalent nature of such a partnership constitutes several challenges for participating entities (Zineldin 2004). For example, power asymmetry between the parties involved poses a great challenge for their partnership. The atmosphere which is characterized by the parties involved as well as their interaction itself is of great importance (Zineldin 2004). Trust is of great importance in this context (Zineldin 2004) and was identified to be among the most important success factors of coopetition relationships (Kwai-Sang et al. 2008). Coopetition is present on digital platforms, especially in the context of two-sided platforms (Ghazawneh and Henfridsson 2013; Tiwana et al. 2010). The aspect of competition has already been discussed. The aspect of cooperation might be seen in the idea to jointly provide a diverse feature set through the digital platform with functional differentiation of the parties involved. While the platform owner is responsible for basic mass-market functions, the third-party developers contribute specialized functionalities. Besides functional differentiation, cooperation can also be seen in the collaboration on software development. The platform owner provides boundary resources, which in turn are used by the third parties to develop and provide applications for the digital platform. As already mentioned, the aspect of power asymmetry in favor of the platform owner is evident in this type of cooperation (Eaton et al. 2015).

Coring systematization

In order to better understand the different versions of coring as well as to locate the research undertaking, classification is a useful tool. Figure 1 depicts various modes of coring while focusing on the special relationship of third-party apps and the platform core in business ecosystems that incorporate a digital application marketplace. Differentiation is done according to the amount of coring and the maintenance of the application. The *amount of coring* refers to whether the whole functionality or only parts are integrated into the platform core. Parts of applications can be interpreted as single functionalities that an application offers to the user. The *maintenance of the application* itself expresses whether the application is still offered in the application store after the coring activity. This is especially relevant, since the involvement of multiple actors in the coring activity might cause conflicts due to opposing interests.

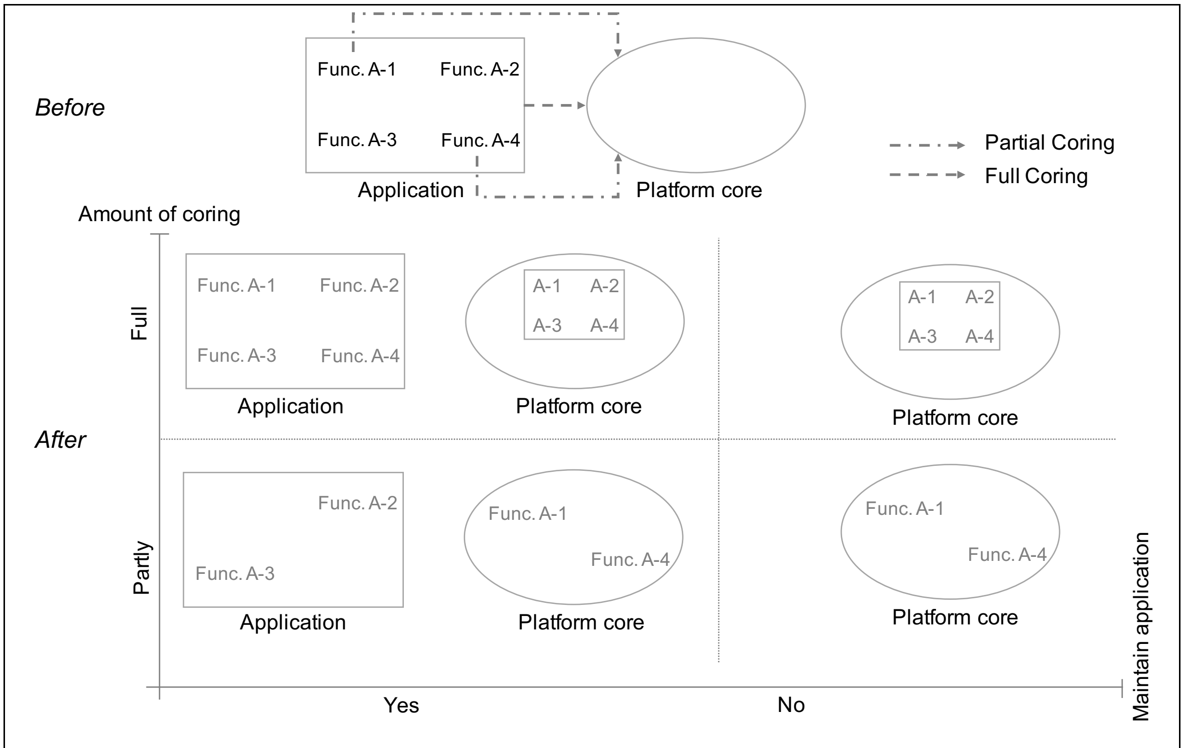

Figure 1. Systematization of coring activities on digital platforms

The upper left quadrant of the systematization includes cases where the full functionality is taken to the core and the application is still maintained and offered in the application store. Examples for such activities are music streaming providers like Spotify or Deezer. The streaming service Spotify started in October 2008, while Apple's similar streaming music service (Apple Music) started in June 2015. With Apple Music, the functionality of music streaming using a subscription model instead of the buy paradigm is available in the platform core for iOS users.

Applications that are integrated into the core but are no longer maintained can be found in the upper right quadrant. Those cases are typically related to acquisitions. A recent example is the application Workflow from DeskConnect, which Apple purchased in early 2017. The formerly paid application was made available for free after the acquisition. Apple announced that they plan to fully integrate the functionalities in upcoming versions of iOS. Though not fully integrated yet, the announcement as well as the free availability can be seen as coring activities.

The lower right quadrant subsumes applications whose functions are partly cored and are no longer maintained. A prominent example for this case is Siri. Most of its functions have been integrated into the core, while the previous application itself was removed from the application store. Before Apple introduced the speech recognition functionality with the iPhone 4s in 2011, the stand-alone app was available in the Appstore and ran on any device. Following the coring, only newer devices could use the iOS built-in function.

The lower left quadrant contains applications that are partly migrated to the core and are still maintained. Examples like the one which will be investigated further within this contribution are located there. Parts of the application are integrated in the platform core, but the application is still maintained. One can further distinguish examples by examining whether or not the cored functions are still provided by the maintained application. Serving as a specific example, both WhatsApp and iMessage offer messaging services over the internet. However, the iMessage functionality integrated in Apple Messages application (part of the core) offers similar, but not equal functionality.

This categorization is only one possibility for systematizing coring activities. Several other criteria can be used for differentiation as well, e.g., taking the issuer of the application into account, which might be of more importance depending on the research focus.

Effects in Different Coring Modes

In the following, the existence of the three related effects for the different coring modes will be discussed. This should serve as a basis to better understand the complex dynamics and related motivations, as well as to identify the coring mode that is able to satisfy multiple stakeholders at the same time and foster platform progress. To accomplish the aforementioned goal, the situation following the (first) coring activity is focused on here.

If the application itself is not maintained, none of the three effects apply regardless of the coring amount. This is due to the fact, that the cored application itself is not available anymore. These cases are therefore not suitable to accomplish the goal of achieving platform progress as a collaborative process of platform owner and third-party developers.

If a full coring is conducted and the application is maintained, some of the effects apply. In this case, the application no longer has a functional differentiation. Regarding competition, this might intensify the effect, since the offerings are closer than before. The coevolution effect is still present and might be somewhat effected by this. This is in contrast to the effect of coopetition, which is heavily influenced by this coring mode. This is especially true for the aspect of cooperation. Since the above-mentioned aspect of separation in the functional offering does not apply anymore, since all functions are available in the core, the part of the cooperation effect is not fulfilled anymore. As a result, the effect of coopetition is no longer fully present. Due to the missing cooperation aspect and the resulting situation that heavily limits the idea of a mutual beneficial partnership, this mode of coring is not seen as able to effectively foster platform progress.

If a partial coring is conducted and the application itself is maintained, all three of the effects apply. The effect of competition applies for the functionality that both solutions provide. The coevolution effect is, like the full coring mode, not affected. The effect of coopetition also applies. In contrast to the full coring mode, the functional differentiation is at least partly maintained, and therefore the cooperation aspect applies. If a specialized category of system functions is cored, this allows third-party developers to develop their functionalities even more effectively and thereby fosters platform progress and the realization of synergies among the participating parties.

The focus now remains on coring activities with functions that are partly migrated to the core while the applications are still maintained, since this mode is believed to be the only one, according to the systematization, in which positive coopetition between the parties could fosters platform progress. The case-study example was chosen to belong to this type of coring.

Case-Study Analysis

The phenomenon of platform coring shall be investigated in the context of two commonly used software components on a digital platform for mobile devices using a case-study approach. The case study as a first step should show whether or not the coring phenomenon is present on digital platforms in the mobile device sector. Therefore the methodology to examine coring activity of a mobile operating system (core) and an innovative application is presented. Afterwards, an appropriate pair for comparison is chosen and shortly described. Finally, the results are presented.

Method

This analysis focuses on the concept of platform coring and more specifically on applications that are partly migrated to the core and still maintained. Thereby, especially the transfer of functions from the specialized applications to the platform core will be analyzed. Both application developers and platform providers regularly release new versions of their applications and platforms, respectively, which typically contain new functions as well as bug fixes.

To examine the coring phenomenon, the following approach is used. In a first step, information about new functions of a selected third-party application as well as the platform core is collected as a basis for comparison. The release notes of the providers serve as reliable sources for the release of new functions. In addition, official blogs are used. The next step involves matching the released functions of both application and platform core. Even though function implementations might not be exactly equal, it is critical to identify features that fulfill a similar goal in either software component. Thereafter, metadata for the relevant functions are gathered. Especially release dates are of great importance, as they indicate where the function was available first and therefore serve as a classification criteria for coring activities. This comparison is conducted for any pair of functions that could be matched between the core and the third-party application. Finally, the results are aggregated and patterns are identified. Additional sources such as customer reviews may be used to double-check the results and thus increase reliability.

Selection of Comparison Objects

Apple iOS is a widely-used operating system (OS) for mobile devices. Since its first release in 2007 (formerly the iPhone OS and iPad OS), iOS is currently available in its 10th version. The OS is issued with any Apple mobile device except the Apple Watch. This study focuses on the functionality of iOS for the smartphone series iPhone. iOS offers various basic functions for using the smartphone. Over the past years, the built-in functions as well as related applications have improved incrementally. By definition, previous research requires platform cores to serve at least one essential function within the system and to be easily connected to or built upon to expand the system (Gawer and Cusumano 2008). Both aspects are fulfilled for iOS. In the context of this research, iOS serves as the connecting element for the participating actors in the business ecosystem. Regarding the second requirement, iOS provides an architecture that can be extended by third-party developers providing applications, which can be distributed using the related AppStore (Ghazawneh and Henfridsson 2015). According to Ghazawneh and Henfridsson (2015), the Apple AppStore can be classified as a censored marketplace. iOS is chosen as the platform core since it fulfills the requirements and has an industry-wide strong position regarding innovation activities (Eaton et al. 2015; Ghazawneh and Henfridsson 2013).

WhatsApp is an internet-based messaging service that allows smartphone users to send text messages and share media files like photos and videos. In 2015, WhatsApp added new functions that allow users to make phone and video live calls. In contrast to the formerly asynchronous communication functions, this added synchronous features to the service portfolio. The application is widely used by smartphone users of various platforms. In 2014, WhatsApp was acquired by Facebook. While WhatsApp had changed its payment model multiple times in the past (single payment, yearly payment, free), it was announced that the service would be free of charge from early 2016 onward. The smartphone application is available for all major mobile operating systems, including iOS. For this study, the focus is on the functionality of the iOS application for iPhones to be consistent with the related platform core.

In order to be well suited for comparison, the two software components need to at least serve similar functions and show progress in their domain. Both aspects are fulfilled when considering the software pair Apple iOS and WhatsApp for iOS. As far as progress goes, both solutions are known for their innovative solutions. A major part of the reduction in use of the short message service (SMS) is attributed to the continuous improvement of WhatsApp (Shambare 2014; Church and Oliveira 2013). Apple is known for their introduction of novel technologies that exhibit a high acceptance rate within a short amount of time (Eaton et al. 2015). The following comparison clarifies the common functions between the software components.

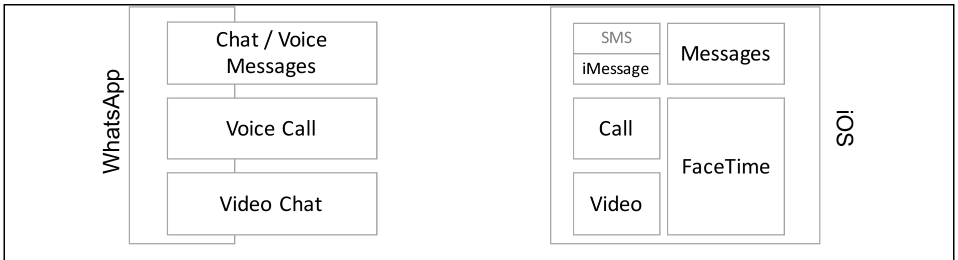

Figure 2. Systematization of coring activities on digital platforms

Figure 2 displays the major function blocks of WhatsApp as well as the corresponding iOS functionalities for internet-based messaging. The functions of WhatsApp have been grouped into three blocks according to similar functions that were found in iOS. The chat function of WhatsApp is also available in iOS, which uses iMessage for internet-based message exchange. Similarities in both services involve the options to send media files and conduct group chats, as well as other asynchronous features like voice message transmission. In contrast to WhatsApp, the message application of iOS combines internet-messaging with traditional SMS capabilities. The two other groups provide synchronous communication features. Voice calls in WhatsApp resemble iOS's FaceTime voice calls. Both allow for direct interaction like traditional phone calls, but through the use of Voice-over-IP (VoIP) transmission. In a similar manner, FaceTime video calls are like WhatsApp video chats, which also use VoIP. Several WhatsApp users recognized similarities to iMessage as well, and expressed those in their application reviews on the AppStore.

Besides the similarities on the functional level, there are many other similarities regarding organizational and usage aspects of the two services that make them well-suited for comparison. Both solutions use phone numbers as unique identification criteria. Therefore, no exchange of user names or additional information is necessary. iMessage additionally allows for the use of an e-mail address as an identification characteristic. Both solutions can be used on multiple devices. iOS messaging features are available on mobile devices of Apple as well as on all newer versions of macOS. However, usage possibilities are restricted to the Apple ecosystem. WhatsApp can be used on multiple mobile devices and different operating systems. The web edition and standalone PC-software allow for the usage of WhatsApp on a computer. Another important similarity is that both solutions are free of charge in their usage. While a wide range of similarities exists, this paragraph only covers a brief selection.

Moreover, there are significant differences between the two solutions that lay the groundwork for this research methodology. First, the service provider constitutes an important difference. While the iOS functionalities are provided by Apple as the system provider, the communication features of WhatsApp are provided by WhatsApp Inc. The other and probably most obvious difference is their availability. iOS messaging features are part of the core and are therefore available by default when using a corresponding device (e.g. an iPhone). WhatsApp features are available only after downloading the corresponding application WhatsApp Messenger. Closely related to that is the aspect of functional diversity and level of integration. Since WhatsApp is a third-party application, it is restricted to the functionality and options provided by the boundary resources of the platform. Additionally, the level of integration with the platform is typically lower for third-party applications than it is for applications from the provider. The platform owner and the functionalities of the core will always have advantages when it comes to integration aspects and freedom of realization (Eaton et al. 2015). This holds especially true for Apple, who is known for their restrictive behavior in relation to other mobile operating system manufacturers and platform owners (Ghazawneh and Henfridsson 2015; Eaton et al. 2015; Tilson et al. 2012).

Since it might be considered critical to compare a platform core with a single application, it is important to highlight similar functions and aims of the software components to provide a solid basis for the analysis. Based on the aforementioned aspects, this is seen as fulfilled for the following analysis. Furthermore, for demonstration purposes regarding the existence of the coring phenomenon on digital platforms, the comparison approach between just two components is an appropriate first step.

Data Sources

In order to compare functionalities of the WhatsApp Messenger and iOS as the platform core, reliable information regarding the two software components is necessary. Therefore, several data sources were combined to fulfill this requirement. These data sources are briefly described in what follows.

The official release notes of iOS were used to track the new functionalities available in the platform core. Since the above-mentioned software packages are issued with the operating system, this approach seems suitable. As far as the WhatsApp Messenger is concerned, gathering release notes was not as straightforward. The Apple application store (AppStore) only provides information for a couple of recent versions of the application. Since information is required since the initial release of the application, additional sources were needed. Therefore, the official WhatsApp blog as well as additional announcements on the web were used to gain further insight. Moreover, several third-party websites tracking the progress from applications over time were used in combination to complement the information regarding WhatsApp features. Application review websites were another useful source, especially in combination with videos reviews and tutorials that demonstrated the application features in a practical manner. Finally, over 130,000 user reviews of the WhatsApp Messenger from the Apple AppStore were used for detailed information and cross-validation of the feature list.

Analysis Results

This section presents the results of the analysis regarding similar functions of the platform core and WhatsApp Messenger. In accordance with the above-mentioned approach, 27 functions were revealed that offer similar functionality across the two software components. Thereafter, the corresponding release dates were gathered for each functionality of the two software components. The combination of the release dates was crucial to classify whether the individual feature had been cored. Table 1 summarizes the corresponding software functions and their release order.

For 25 out of 27 functional aspects, a first mover could be determined. For the two remaining functions (camera instant button and report spam) the function release dates are both within a two-week period, so that no first mover could be defined. With respect to release dates of the first two functions, a restriction had to be made since no exact release date could be identified. Therefore, the dates in the table were based on an application review where those functions definitely existed, although they might have been available earlier. Another remark needs to be made for the keyword search functionality in iOS. In fact, this function had already existed in the previous messaging application of iOS before iMessage was even introduced and combined with the traditional SMS capability, meaning that the release date of the function itself had been before the iMessage launch.

For the 25 functions for which a clear first mover could be determined, the vast majority were initially available on WhatsApp. 19 out of 25 (76%) functionalities were first available in the WhatsApp application for iOS. In contrast, 6 out of 25 (24%) had first been available in the platform core and were later added to the WhatsApp messenger application. The result becomes even clearer if only asynchronous messaging functionalities are considered. This seems reasonable since WhatsApp started a messaging tool which provides cross-platform messaging, whereas iOS and its FaceTime functionality is a fully featured VoIP application. In this case, from the asynchronous features (audio and video calls excluded) 19 out of 23 (82.6%) of the functionalities were first available in WhatsApp and only 4 out of 23 (17.4%) were first available in Apple's iOS.

Another aspect to be considered is the time for which the functions were unique. The two in the sample case are very similar. The average time from releasing a function until a similar function was available in the corresponding software is slightly above two years for both solutions if all functions are taken into account. The average time if Apple's iOS is the first mover is around 770 days, while if WhatsApp is the first mover

Function	WhatsApp	iOS	Innovator
Indication when chat partner is typing	10/2009*	10/2011	WhatsApp
Profile pictures in chat overview list	10/2009*	09/2015	WhatsApp
Show message send time	11/2009	09/2013	WhatsApp
Message delivery indication	12/2009	11/2010	WhatsApp
Send current location	02/2010	11/2010	WhatsApp
Send contact information	02/2010	11/2010	WhatsApp
Block contacts	05/2010	09/2013	WhatsApp
Forward and delete messages from a chat	06/2010	10/2011	WhatsApp
Keyword search for all chats	11/2010	06/2010**	iOS
Group chats functionality	02/2011	10/2011	WhatsApp
Title for group chats	02/2011	09/2014	WhatsApp
Media browser for overview in chats	04/2011	09/2014	WhatsApp
Mute individual group chats	09/2011	09/2014	WhatsApp
Group owner can drop members	07/2012	09/2014	WhatsApp
Message encryption	08/2012	10/2011	iOS
Send multiple photos at once	07/2013	09/2014	WhatsApp
Voice messages	08/2013	09/2014	WhatsApp
Modify images before sending (e.g. crop)	12/2013	09/2015	WhatsApp
Modify videos before sending	09/2014	09/2015	WhatsApp
Camera instant button	09/2014	09/2014	-
Message read receipts	11/2014	10/2011	iOS
Quick photo access (recent photos)	02/2015	09/2014	iOS
Voice calls over IP	04/2015	09/2013	iOS
Report spam	04/2015	04/2015	-
Rich link preview in chat	11/2015	09/2016	WhatsApp
Send single emojis bigger	07/2016	09/2016	WhatsApp
Video calls over IP	11/2016	06/2010	iOS

Table 1. Release date comparison of similar software functions in WhatsApp and Apple's iOS

* = Exact release date could not be determined. At the specified date the function definitely existed, even though it might have existed before.

** = Keyword search functionality was available before iMessage launch, since the function is part of the core messaging application which combines SMS and iMessage.

the average time is 781 days. When taking a similar perspective with a focus on asynchronous communication features, the variation in the average time between the two groups increases. In this case, the average time for iOS as a first mover decreases to 425 days, while the corresponding 781 days for WhatsApp remain. In summary, when WhatsApp released similar functions later, it took slightly over one year until their launch, while for iOS the average time was two years. A closer look reveals some structural inequalities with respect to the distribution, so that the average amount is not suitable as a single measure. Figure 3 shows the distribution of asynchronous functions transferred to the other platform for different time frames. The figure reveals that when iOS is the first mover, the availability of similar functions in WhatsApp is quite faster than the other way around. Most functions in WhatsApp that previously existed in iOS are available within one year, while the majority of functions in iOS that previously existed in WhatsApp are available after at least one year.

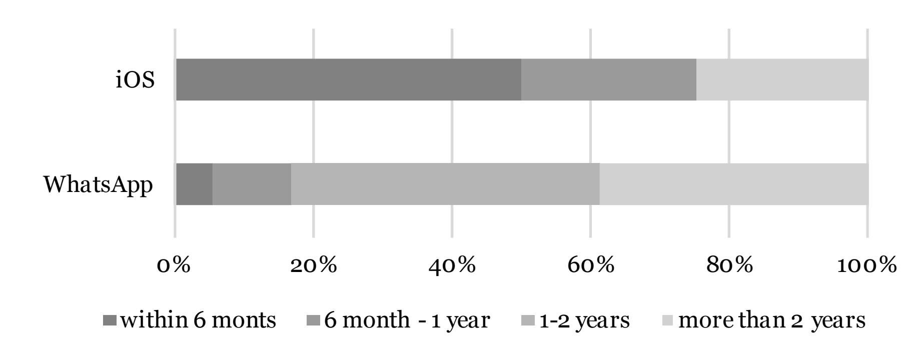

Figure 3. Time gap histogram until similar asynchronous functionalities were available in corresponding software component (per first mover)

Coring fundamentals

The following section discusses key aspects of coring on digital platforms, while focusing on the case of partly migrated functions for which applications are maintained. Central aspects such as relevant parties, their motivations, and related chances and risks are highlighted using the example of WhatsApp and iOS. Further, specifics of the case are discussed which enable first attempts at generalizing the structural aspects.

Involved Parties

In the context of platform coring, two types of parties are relevant. First, the group of third-party developers who develop applications for the corresponding digital platform and therefore provide functionalities that could be cored. Since functionalities are not exclusive to a single application on a platform, this group might contain various third-party developers depending on the individual case. If coring is interpreted as a directed transfer, the group of third-party developers is the source, but not the issuer, of coring.

Second, the platform provider that provides the platform core could be seen as the destination of the coring activity. The platform owner provides the platform (core) itself, as well as the boundary resources that allow the third-party developer to develop applications for the platform. The platform owner typically provides regular updates of the core with enhanced functionality. If the new functionalities incorporate functions that have previously existed in third-party applications, the phenomenon of coring is present. The platform provider is the party that executes the coring activity, since it is the only one responsible for and able to modify the platform core.

In the demonstration case, Apple iOS serves as the platform core, while WhatsApp serves as the source of coring. Even though it needs to be acknowledged that the group of third-party developers (potential sources) might include more applications (e.g. other messaging applications), the example pair serves to demonstrate that the phenomenon exists. iMessage itself was introduced to iOS with a regular update.

Aims of the Involved Parties

The *platform owner* is the linking entity between platform users and the third-party developers (contributors), who utilizes the platform core for this purpose. The platform owner strives to offer a highly-valued platform to both sides. The value of two-sided platforms could be separated into value for complementors and users. Due to the underlying platform dynamics, it is crucial for platform success and survival that both sides engage on the platform (Ghazawneh and Henfridsson 2013).

Regarding the user side, the platform owner tries to offer an attractive digital platform. Thereby, users might value the platform's core functions as well as the functions from complementors' applications. An attractive platform core should include commonly used functions for the mass-market. In many cases, the platform core is useful, even without complementors' products (Haile and Altmann 2016). One implication of platform coring for users is the provision of more functionality of a digital platform by default. Since functions that formerly were provided by third-party applications are available in the core afterwards, the platform core provides more value to the user, even though the total potential value, as the functionality combined of core and third-party applications, might not have changed. Core functions are usually easier to access than functions provided by third-party applications. Since the core typically provides functions that are used by the mass-market and since functional requirements and usage patterns develop over time, the provision of commonly used functions is useful for customers.

Besides the value of core functions, users will value the third-party applications offered on a platform since they provide enhanced functionality (Haile and Altmann 2016). The more complementors are active on a digital platform, the more applications will be available, which in turn increases the value provided to the user of the platform (Boudreau 2012). It is therefore critical for the platform owner to attract complementors. For them, the platform should provide a framework that allows them to efficiently develop their specialized functions as applications for the platform (Hsieh and Hsieh 2013). This includes the provision of boundary resources as well as common system functions that are needed by third-party applications.

The *third-party developers* on digital platforms aim for customers of the platform core (all users) to become their users as well. It is therefore essential for the complementors to provide value to the platform customers which the platform itself does not provide. Typically, this involves functionalities for specialized contexts that the platform core itself does not provide (Haile and Altmann 2016). From the complementors' perspective, applications that are solely provided for one platform could be differentiated from applications that are provided for multiple platforms. In the case of cross-platform applications, especially when network effects are present, the complementor is interested in having as many users from various platforms as possible to provide high value to the customer within the ecosystem across various digital platforms (Hyrynsalmi et al. 2016). If the application is only available for a single platform, the complementor is also interested in having many users, but the success of the total ecosystem does not depend so much on it as it does in the cross-platform case. For applications for that, network effects apply and are available for a single platform; having many users is important as well.

The implications of coring for third-party developers are quite different from those from the user perspective. To understand the implications, one might differentiate between application functions and system or platform functions. System or platform functions are functions that can be used to build applications upon. One example is an encrypted storage for application data. While the developers initially needed to develop the function themselves, a newer version of the core might contain such a function, which could then be used by third-party developers. Application functions are functions which customers use directly. They provide direct functional value to the user. In general, functions that are cored and are still available in third-party applications provide no more functional differentiation. While the substitution through coring can easily be realized for simple functions like a flashlight application, the case of more advanced functions is more complex. For example, aspects like function bundling or network effects highlight the importance of not considering the functions in isolation.

In the demonstration case, Apple as the platform owner strives to provide a competitive platform (core) that is attractive to users as well as complementors. Since the mobile device sector encompasses various competing platforms, this is important for Apple (Hsieh and Hsieh 2013). From the user perspective, Apple wants to appear as an attractive platform that provides basic, commonly used functions by default and offers additional applications, so that specialized functionality can be gathered using the platform's marketplace. Another user-related motivation for introducing iMessage might be the provision of an ecosystem-wide messaging service. For third-party developers, Apple also wants to be an attractive platform which provides them with attractive boundary resources and development resources, so that they develop applications for the platform. WhatsApp has a huge interest in being available on multiple platforms, for example due to the network effects discussed above. Furthermore, WhatsApp wants to provide a modern communication service and customer retention even across platforms. For Apple, this is important as well, since the availability of common applications that might have been used before makes the migration to the platform less complicated for users. In general, of course both parties are interested in gaining users.

Using the example of iMessage and WhatsApp, it becomes clear that the transfer of WhatsApp functions to the core cannot fully substitute the value WhatsApp provides to the user in either case. This is, for example, due to network effects that apply for messaging services like these two. If a function is available to the user but cannot be used in combination with the desired recipient, the value diminishes. WhatsApp is available on various mobile platforms, while iMessage is only available within the Apple ecosystem. Another example is functional bundling as the combination of a set of functions available in the WhatsApp application. If a user wants to send an image to a group of people, it is good if the application has a group feature as well as an image sending feature. If the group feature is missing, the user needs to send the image to each desired recipient separately.

Structural Inequalities in the Partnership

As previously highlighted, the existence of coopetition poses a challenge for all parties involved. This is especially true since major structural inequalities exist between the positions of the platform owner and the complementors on digital platforms in the context of coring. Power asymmetry is known to be a major challenge in the context of coopetition relationships (Zineldin 2004).

The platform owner as the issuer has full control over the platform. This applies to any technical aspects as well as the platform governance. The owner is therefore able to change processes, architectures, prices and many other aspects related to the platform. Furthermore, the owner provides the boundary resources for third parties to develop applications.

The third-party developers use the platform resources as well as the boundary resources to develop applications for the platform. To do this, they need to conform to the rules and guidelines set by the platform owner in order to be able to publish their application on the platform. In addition, some digital platforms incorporate a review process for third-party applications that must be passed prior to their release on the platform's marketplace.

Even though the platform owner has a tremendous advantage regarding power in the relationship, it is important for the owner to satisfy both sides of the platform (Ghazawneh and Henfridsson 2013). This is especially true if competition exists on the platform / ecosystem level. The dependence on the complementors' activity might be the major restriction on the platform owner's behavior regarding changes on the platform and for coring in general.

In the demonstration example, Apple as the platform owner provides boundary resources as well as an SDK for third-party developers. WhatsApp uses these resources to develop their application. Furthermore, WhatsApp conforms to the application guidelines, since otherwise the publication of applications would be restricted by Apple during the application review process, which already happened and forced WhatsApp to conduct the required changes before the new application was released, pertaining to the WhatsApp blog.

Opportunities of Coring

For *platform providers,* coring is a chance to provide enhanced value to the users through the platform core itself. The more functions the core offers, the more functional value the core provides. A state of the art two-sided digital platform needs to attract both users and developers to participate on the platform. Another potential of coring is that the more functions are cored, the less dependent the platform owner is on complementors to provide an attractive platform. Coring in this case is also a risk-mitigation strategy for the platform owner. Furthermore, coring provides another possibility of enhancing control over the platform, since the platform owner alone is in charge of coring functions. In contrast to third-party functions, where general rules and guidelines can be set, the actual implementation cannot be controlled. Platform coring furthermore ensures platform hygiene, since redundant applications can be removed through this procedure. Regarding the complementors, coring allows the platform owner to provide useful system functions for third-party developers to use for their application development. Coring can therefore help to increase platform attractiveness for complementors. On the other hand, the provision of system functions allows the platform owner to realize lock-in effects for complementors. Realizing similar functionality on another digital platform that does not offer a similar basic set of system functions would require the developer to invest additional effort to realize the basic functions on its own. Finally, coring can be seen as a chance to generate more revenue. Even though the vast majority of core functions are free, the platform owner could charge users for using specific core functions or exploit usage data (e.g. interests for advertising). Coring would therefore shift the revenue from the complementor to the platform owner. Another opportunity for platform owners is the use of third-party applications as a market test instrument. Coring is then applied only to functions that are frequently used by many and are therefore relevant for the mass market.

For *third-party developers,* coring results in the possibility of having a feature-rich digital platform as a basis for their development. Having many platform system functions already in place reduces their effort needed to develop platform applications and furthermore facilitates adherance to platform standards. Digital platforms that provide this are a good basis for third parties to develop complementary applications.

Risks of Coring

The main risk associated with coring activities for the *platform owner* is the potential demotivation of third-party developers. Depending on the type of functions and the extent to which coring is conducted, the additional value through third-party applications and their functions diminishes. As already discussed earlier, the continued participation of third-party developers is crucial for the success of the digital platform. When developers stop developing for a digital platform, their overall attractiveness declines, which poses a threat for the platform owner (Tiwana 2015). A second risk of coring for the platform owner is the platform maintenance due to increasing platform complexity. The more functions are available in the core, the bigger and more complex the core becomes. Through the increased amount of code and dependencies, the related effort increases. In general, the risks associated with coring for the platform provider are controllable, since the platform owner is the party that both initiates and conducts coring activities.

For *third-party developers*, several risks are associated with coring. The first and most severe risk is that of losing functional differentiation. If functionality that was initially provided by complementors' applications is then provided by the platform core, the right to exist to extend basic platform functions does not apply anymore. From a user's perspective, this might result in ceasing to use the application and the number of new users might decline as well. Since using third-party applications requires additional effort in contrast to built-in functions and potential charges may apply, it is likely that users will tend to use the core functions. Over time, it is likely that the overall number of customers diminishes. Nonetheless, it should be remarked that several restrictions apply. Several factors besides functional value influence the application value and users' motivation to use an application. The transfer of functions itself might not be sufficient to make users stop using an application. Other aspects such as cross-platform availability and network effects serve as examples. Nevertheless, a revenue decline might be the result if users migrate. Since the development and maintenance of applications requires resources, further advancements might be limited. Furthermore, it should be highlighted that typically no effective governance mechanisms for complementors exist to prevent the platform owner from coring. Another aspect is that developing applications on a digital platform with a rich feature set might create lock-in effects (this corresponds to an opportunity for the platform owner). For complementors, the realization of a similar application on another platform requires additional effort, since the basic functions might not be available on other platforms.

Taken together, while for the platform provider the opportunities outweigh the risks, the third-party developer faces risks which constitute an unknown component and might have truly negative consequences. This is emphasized by the fact that there are typically no direct countermeasures to protect them against the negative consequences of coring. The power asymmetry for the platform owner is quite present in this partnership.

Coring as Beneficial Cooperation for Platform Advancement

Combining the fundamentals, the opportunities and risks, as well as the dynamic aspects mentioned before, the question arises as to what a beneficial partnership for both parties might look like. The following section will briefly elaborate on this with a focus on the functional aspects of digital platforms.

Regarding the functional aspects, once again, the separation between system / platform functions and user functions must be highlighted. System functions are used by complementors to build their applications upon and can be seen as part of the boundary resources or at least of the software development kit provided by the platform owner. Regarding those components, coring is assumed to be viewed very positively by third-party developers, since their availability allows them to effectively develop their specialized applications. A feature-rich platform core could therefore attract complementors to the platform and constitute a differentiating factor among competing platforms (Hsieh and Hsieh 2013). Even though these system functions do not provide value for the platform user directly, they contribute to the platform uniformity and professionality in the sense of software modularity, and therefore provide indirect value to the user.

For the user functions, the positive perception of coring might not apply. Quite to the contrary, coring might be viewed very critically by complementors. Since the user functions are directly accessed by the user, they have a great influence on their value perception. If those are cored, the functional differentiation of the third-party applications diminishes or is lost if all functions are cored. Since this constitutes a severe threat for the developer, it is essential to maintain a functional differentiation between the core and the third-party application in order to ensure complementors' motivation to participate on the platform. This conforms to the general idea of the core and its extension through specialized applications. Even though the right to exist includes quite more than just functional differentiation (e.g. value due to network effects), over the long term these functional aspects play a vital role. While time progresses and user requirements change, it becomes necessary for the platform owner to provide additional common functions in the core, even though it might hurt some complementors, in order to stay competitive. It is difficult yet important for the platform owner to make well thought out decisions in this area of conflict to ensure platform progress while maintaining continued participation of users as well as complementors. Finding the right balance between complementors', users' and platform owners' interest is an ongoing task associated with the management of platforms.

Taken together, the use of coring to establish a symbiotic relationship between platform owner and third-party developer should predominantly focus on system functions. The coring of user functions should only be conducted to the extent that it accelerates general platform progress, but should not be controlled by expected revenue gains, since this is assumed not to work in the long run. The aforementioned considerations solely focus on platform coring; this should in no way prevent the platform owner from introducing new innovations to its platform.

Discussion

Altogether, the case study results clearly demonstrate the existence of the coring phenomenon on digital platforms for mobile devices. Various similar messaging functions have been integrated in the platform core over time. Additional negative consequences, limitations and future research are discussed below.

The theoretical considerations along with the case-study example showed that the coring of user functions reduces functional differentiation and therefore could pose a strategic threat to third-party providers such as WhatsApp. This should not mean that WhatsApp has no benefits left in comparison to the messaging functionality of iOS, especially since the application provides its services across multiple platforms, which is also recognized as an advantage in customer reviews. This is consistent with Selander et al.'s (2013) suggestion that partners should innovate within multiple ecosystems simultaneously, and the risk management strategy idea of Hyrynsalmi et al. (2016). To be active in multiple platforms reduces complementors' dependence on a single authority and also provides access to additional customers.

Due to the structural inequalities in the partnership between platform owner and complementors, a major power asymmetry exists. Even though the power advantage of the platform owner can be used to effectively guide platform advancement and ensure platform hygiene, negative examples exist as well. In censored marketplaces like the Apple AppStore (Ghazawneh and Henfridsson 2015), contributed applications as well as updated versions are reviewed and become available only after approval by the owner. Criticism related to the review process encompasses functions and design aspects that formerly existed in third-party solutions and were integrated into the platform core thereafter. A severe example is the iTunes wireless sync function. According to several sources, Apple used their power as a platform owner and declined an app providing wireless synchronization functionality from the AppStore. Since one of the next major iOS releases, a function like the one previously provided by the declined app has become available in iOS itself (Goodin 2011). Similar examples involve other user functions like the control center in newer iOS versions. As mentioned before, it is difficult yet important for the platform owner to make decisions regarding coring activities carefully, to ensure continued participation of users and complementors while pushing the platform forward.

Limitations

Concerning the analysis approach and data analysis, several limitations need to be considered. First, it should be remarked that the selection of functions depends on the researchers' interpretation. This is especially true for the matching aspect of similar functions across the two software packages. Since no weighting was performed on the functions themselves, they were all seen as equally important. However, future studies should incorporate this aspect, as the value provided by the different functions as well as their implementation effort varies. Furthermore, the possibility of path-dependencies between the functions was not examined. Since some functionalities constitute preconditions for other functions, this aspect should also be taken into consideration. Due to data availability and the approach of this study, only user functions have been considered. As already mentioned, time difference is a critical measure in this context that provides useful insights. Consider, for example, the release policy of both companies. While new iOS functionalities are bound to system updates only, WhatsApp versions are released both with iOS system updates and independently on an irregular basis. Moreover, as WhatsApp is a multi-platform application, the past has shown that the release of new functions is not synchronous for all platforms. Therefore, one could argue that, due to the high regulation standards and the review process of Apple, iOS functionalities are released later than on other platforms like Android or Blackberry, as several examples prove. Finally, it should be considered that the study only used public release dates. Neither Beta tests nor announcements were considered, which might happen to play a role as a source of pre-coring innovation processes for the other party. Furthermore, functionalities that are due to organizational advantages were neglected. One prominent example are interactive responses from the control center or push notifications, which were first available in iOS when Apple introduced the option. Of course, WhatsApp was the second mover, since the related developer resources were released later for third-party developers. Regarding the quality of data sources, this study aimed at using as many official sources as a possible. However, since the availability of information for different time frames varied, additional data sources as well as secondary sources (e.g., reviews) were used to cover as many functions as possible.

Regarding possible inferences, it should be noted that, for the sake of conciseness, this study only focused on a single, though prominent, application out of a vast number of apps in the platform store. In fact, several other applications and platforms bear great potential to be equally relevant for this study. The phenomenon of coring as mentioned earlier could originate from all applications available on the digital platform. It is therefore possible that many more functions, even those not classified as coring in the case study, have been cored from other applications on the platform.

Concerning the fundamentals, it should be noted that they are developed theoretically, even though many aspects are shown within the case study. Nonetheless, the results should be handled with care, especially regarding their completeness as well as the generalized aspects discussed.

Future Research

Regarding the case study, a first idea is to re-conduct the analysis of this contribution in a more detailed manner. Possible extensions include the integration of different function categories, the introduction of performing function weightings, or the consideration of path-dependencies for a more detailed view.

Due to data availability and the approach of this study, only user functions were considered. The differences between user and system functions regarding coring and especially their implications for the participating entities were addressed in this contribution. Future studies could profit from incorporating system functions and therefore provide a more fine-grained view of the coring progress.

While this study used a case study to approach the coring phenomenon, future studies could focus on theory validation using the extensive data the platforms offer, even though they are hardly available. In combination with the idea of integrating system functions in the analysis, this would allow for the developed idea of beneficial coring for effective platform advancement.

In this contribution, a one-core to one-application case study was conducted. For a better and more detailed understanding of coring, it is necessary to consider all applications that might be relevant for coring on a platform. And even more than that, for an overall picture all potentially relevant applications from various platforms should be considered. This would allow the identification of cross-coring initiatives. Even though cross-platform coring activities do not fit the exact definition of coring, the results of cross-platform function transfer are of great importance to better understand coring and innovation dynamics on a multiple platform level. Especially when competition, and potentially coopetition, exist on a platform level, such analysis might contribute to a better understanding of the complex phenomenon and the related effects on a platform level in contrast to the within-platform focus as in this contribution.

Outlook

With the most recent major update of iOS, Apple introduced a new option for partners to add to the core's messaging application. In contrast to the typical application store, the new iMessages-Apps allows third parties to contribute specialized functions within the messaging application. This enables developers to contribute specialized content and functions without having to develop a stand-alone messaging application. Furthermore, this provides access to iMessage's tremendous user base. For Apple, the concept allows them to profit from the innovation power of third parties within their application. Apple thereby fosters innovation within the specialized messaging application and once again maintains the control through boundary resources. Additionally, the platform owner can convert external innovation efforts to increase both customer value and expand their user base, which in turn has a positive impact on customer value. Furthermore, a stronger incentive for third parties to develop extensions for the iMessage application simultaneously decreases their motivation to maintain additional and stand-alone applications. By using this concept, the platform owner can get multiple third-party developers to develop for the core's messaging application at the same time and therefore provide enhanced customer value. Since developers can fully concentrate on their respective fragment, the platform owner is able to aggregate several specialized and highly matured functions into the core's messaging application. To some extent, this concept fits the idea of effective coring for platform advancements. With iMessage-Apps, many system functions that could be interpreted as a little platform itself are provided as a basis for third-party development.

The platform owner now fully controls the dynamics and incentives to develop components for their specific core application – a concept that could be interpreted as *coring by design*. Since the developed components are only accessible and useful in combination with the specific core, the lock-in effect for developers is strengthened. Apple thereby prevents developers from multi-homing as a risk-mitigation strategy.

Conclusion

Today's mobile devices, like smartphones, are part of powerful business ecosystems, which usually involve digital platforms. Those digital platforms and the related marketplaces allow users to extend their device's functionality by downloading additional applications. Through the provision of boundary resources, platform owners enable third parties to develop additional software packages for their platform. In this context, innovation is a critical success for all entities involved. To better understand the evolution of platform core and third-party applications, this paper examined the phenomenon of coring using the example of Apple's iOS and WhatsApp. Specifically, activities regarding platform coring as the integration of several functionalities provided by third-party applications in the platform core were investigated.

The paper starts with a summary of the relevant literature regarding coring on digital platforms, which revealed that slightly different interpretations are present. The paper makes three major contributions to the existing literature. First, through the integration of existing literature, a *systematization of coring activities* has been developed. The coring modes are separated by the amount of coring, as the extent to which functions are transferred to the core, and the application maintenance, as whether or not the specific third-party application is still available after the coring process. Besides elaborating on the study's main case in terms of the systematization, examples have been given for all modes. Second, the *case study* revealed that the phenomenon of platform coring is present on digital platforms for mobile devices. The case study focused on messaging functionalities of iOS as the platform core and WhatsApp as a commonly used messaging application. Similar functions in both solutions were analyzed. More than three-quarters of the similar functionalities identified were first available in WhatsApp and could therefore be classified as coring. Third, the paper aimed to contribute to the body of knowledge by presenting *fundamentals of the coring phenomenon,* including the different aims and structural inequality between the parties involved, as well as risks and opportunities. Thereby, the focus was on partially cored applications that are maintained. Even though coring constitutes a potential threat for third-party developers regarding their functional differentiation, the paper developed an idea for how a beneficial partnership incorporating coring activities might work for both the owner and third-party developer.

References

Antero, M. C., and Bjørn-Andersen, N. 2013. "Why a Partner Ecosystem Results in Superior Value: A Comparative Analysis of the Business Models of Two ERP Vendors," *Information Resources Management Journal* (26:1), pp. 12-24.

Armstrong, M., and Wright, J. 2007. "Two-Sided Markets, Competitive Bottlenecks and Exclusive Contracts," *Economic Theory* (32:2), pp. 353-380.

Benlian, A., Hilkert, D., and Hess, T. 2015. "How Open Is This Platform? The Meaning and Measurement of Platform Openness from the Complementors' Perspective," *Journal of Information Technology* (30:3), pp. 209-228.

Boudreau, K. 2010. "Open Platform Strategies and Innovation: Granting Access Vs. Devolving Control," *Management Science* (56:10), pp. 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), pp. 1409-1427.

Ceccagnoli, M., Forman, C., Huang, P., and Wu, D. J. 2012. "Cocreation of Value in a Platform Ecosystem: The Case of Enterprise Software," *MIS Quarterly* (36:1), pp. 263-290.

Church, K., and Oliveira, R. d. 2013. "What's up with Whatsapp?: Comparing Mobile Instant Messaging Behaviors with Traditional SMS," in: *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*. Munich, Germany: ACM, pp. 352-361.

Darwin, C. 1859. On the Origin of Species by Means of Natural Selection. J. Murray.

Eaton, B., Elaluf-Calderwood, S., and Sorensen, C. 2015. "Distributed Tuning of Boundary Resources: The Case of Apple's Ios Service System," *MIS Quarterly* (39:1), pp. 217-243.

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

Gawer, A., and Cusumano, M. A. 2008. "How Companies Become Platform Leaders," *MIT Sloan management review* (49:2), pp. 28-38.

Gawer, A., and Cusumano, M. A. 2014. "Industry Platforms and Ecosystem Innovation," *Journal of Product Innovation Management* (31:3), pp. 417-433.

Ghazawneh, A., and Henfridsson, O. 2013. "Balancing Platform Control and External Contribution in Third-Party Development: The Boundary Resources Model," *Information Systems Journal* (23:2), pp. 173-192.

Ghazawneh, A., and Henfridsson, O. 2015. "A Paradigmatic Analysis of Digital Application Marketplaces," *Journal of Information Technology* (30:3), pp. 198-208.

Goldbach, T., and Benlian, A. 2015. "Understanding Informal Control Modes on Software Platforms – the Mediating Role of Third-Party Developers' Intrinsic Motivation," in: *Proceedings of the 36th International Conference on Information Systems,* T. Carte, A. Heinzl and C. Urquhart (eds.). Fort Worth.

Goodin, D. 2011. "Apple Pilfers Rips Off Student's Rejected Iphone App." Retrieved 02/03/2017, from http://www.theregister.co.uk/2011/06/08/apple_copies_rejected_app/

Haile, N., and Altmann, J. 2016. "Structural Analysis of Value Creation in Software Service Platforms," *Electronic Markets* (26:2), pp. 129-142.

Hsieh, J.-K., and Hsieh, Y.-C. 2013. "Appealing to Internet-Based Freelance Developers in Smartphone Application Marketplaces," *International Journal of Information Management* (33:2), pp. 308-317.

Hyrynsalmi, S., Suominen, A., and Mäntymäki, M. 2016. "The Influence of Developer Multi-Homing on Competition between Software Ecosystems," *Journal of Systems & Software* (111), pp. 119-127.

Kankanhalli, A., Ye, H. J., and Teo, H. H. 2015. "Comparing Potential and Actual Innovators: An Empirical Study of Mobile Data Services Innovation," *MIS Quarterly* (39:3), pp. 667-682.

Kenney, M., and Pon, B. 2011. "Structuring the Smartphone Industry: Is the Mobile Internet Os Platform the Key?," *Journal of Industry, Competition and Trade* (11:3), pp. 239-261.

Koch, S., and Bierbamer, M. 2016. "Opening Your Product: Impact of User Innovations and Their Distribution Platform on Video Game Success," *Electronic Markets* (26:4), pp. 357-368.

Kwai-Sang, C., Boris, L. C., and Ping-Kit, L. 2008. "Identifying and Prioritizing Critical Success Factors for Coopetition Strategy," *Industrial Management & Data Systems* (108:4), pp. 437-454.

Mohagheghzadeh, A., and Svahn, F. 2016. "Shifting Design Capability to Third-Party Developers: An Affordance Perspective on Platform Boundary Resources," in: *Proceedings of the 22nd Americas Conference on Information Systems*. San Diego.

Nalebuff, B., J., and Brandenburger, A., M. 1997. "Co-Opetition: Competitive and Cooperative Business Strategies for the Digital Economy," *Strategy & Leadership* (25:6), pp. 28-33.

Mosad, Z. 2004. "Co-Opetition: The Organisation of the Future," *Marketing Intelligence & Planning* (22:7), pp. 780-790.

Oh, J., Koh, B., and Raghunathan, S. 2015. "Value Appropriation between the Platform Provider and App Developers in Mobile Platform Mediated Networks," *Journal of Information Technology* (30:3), pp. 245-259.

Rai, A., and Tang, X. 2014. "Research Commentary—Information Technology-Enabled Business Models: A Conceptual Framework and a Coevolution Perspective for Future Research," *Information Systems Research* (25:1), pp. 1-14.

Saarikko, T. 2016. "Platform Provider by Accident a Case Study of Digital Platform Coring," *Business & Information Systems Engineering* (58:3), pp. 177-191.

Selander, L., Henfridsson, O., and Svahn, F. 2013. "Capability Search and Redeem across Digital Ecosystems," *Journal of Information Technology* (28:3), pp. 183-197.

Shambare, R. 2014. "The Adoption of Whatsapp: Breaking the Vicious Cycle of Technological Poverty in South Africa," *Journal of economics and behavioral studies : JEBS* (6:7, (7)), pp. 542-550.

Tilson, D., Sorensen, C., and Lyytinen, K. 2012. "Change and Control Paradoxes in Mobile Infrastructure Innovation: The Android and iOS Mobile Operating Systems Cases," *2012 45th Hawaii International Conference on System Sciences*, pp. 1324-1333.

Tiwana, A. 2015. "Platform Desertion by App Developers," *Journal of Management Information Systems* (32:4), pp. 40-77.

Tiwana, A., Konsynski, B., and Bush, A. A. 2010. "Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics," *Information Systems Research* (21:4), pp. 675-687.

Toppenberg, G., Henningsson, S., and Eaton, B. 2016. "Reinventing the Platform Core through Acquisition: A Case Study," in: *2016 49th Hawaii International Conference on System Sciences (HICSS)*. pp. 4634-4643.

Um, S., and Yoo, Y. 2016. "The Co-Evolution of Digital Ecosystems," in: *Proceedings of the 37th International Conference on Information Systems,* P. Ågerfalk, N. Levina and S.S. Kien (eds.). Dublin.

Um, S., Yoo, Y., Wattal, S., Kulathinal, R., and Zhang, B. 2013. "The Architecture of Generativity in a Digital Ecosystem: A Network Biology Perspective," in: *Proceedings of the 34th International Conference on Information Systems,* R. Baskerville and M. Chau (eds.).

Wong, T. Y. T., Peko, G., Sundaram, D., and Piramuthu, S. 2016. "Mobile Environments and Innovation Co-Creation Processes & Ecosystems," *Information & Management* (53:3), pp. 336-344.