The Dynamics of Digital Platform Innovation:
Apple's Strategy to Control Modular and Architectural Innovation in iOS

Ben Eaton
Copenhagen Business School,
Department of IT Management,
Copenhagen, Denmark.
&
Westerdals Oslo ACT,
Department of Technology/IT,
Oslo, Norway.
be.itm@cbs.dk

Abstract

Curated platforms provide an architectural basis for third parties to develop platform complements and for platform owners to control their implementation as a form of open innovation. The refusal to implement complements as innovations can cause tension between platform owners and developers. The dynamic concerning this control of innovation is not well understood in platform literature. This research attempts to address that gap by using qualitative methods to build narrative networks which analyze 45 examples of contested platform innovation. This approach, informed by empirical data sourced from 4664 blogs, identifies patterned sequences of actions describing tussles across the examples. Mechanisms are then identified, which explain how control is asserted and resisted. The principle contribution of this research is to describe and explain the dynamics of contested innovation on curated digital platforms. In doing so, it uses IS notions of digitalization to challenge traditional understandings of innovation in platform architectures.

1. Introduction

Mobile technology platforms, such as iOS and Android, are a locus for the third party innovation of digital services as apps, or executable pieces of software code, offered to end-users [1, 2]. In this way mobile platforms are a type of industry platform, defined as:

"building blocks (they can be products, services or technologies) that act as a foundation upon which an array of firms (sometimes called a business ecosystem) can develop complementary products, technologies or services." [3, p.45].

Digital platforms, such as Apples iOS, have proved to be highly disruptive of established markets, such as the former walled gardens of the mobile operators [4]. Until recently, most academic research into platforms was confined to scholars concerned with product development, technology strategy and industrial economics [5] who together have theorized platforms in the following way. Architectural notions of platforms focus on modular components that can be selectively used and reused across multiple platform complements [6]. The platform owner sits at the nexus of an ecosystem where it possesses core functionality [7], and mediates consumers and developers in a two sided market [8], which is subject to network externalities [9]. The platform owner has property rights over its core functionality [10], which enables it to profit from transactions that it enables. Furthermore, it has the power to act as a public regulator [11] allowing it to maintain the health of its platform ecosystem and to open and close its ecosystem to third party developers of (platform) complements [6]. Finally, platform scholars [3] acknowledge that platform owners and third parties engage in the practice of open innovation [12, 13] as they innovate in and around a platform.

These perspectives of platforms are informed by concepts of hardware [14]. In contrast, Information Systems (IS) interest in platforms has a greater focus on software and its corresponding digital characteristics [15]. Within IS, platforms are considered to be a form of digital infrastructure [2], and are organized as layered modular architectures [16], which provides possibilities for both enabling and constraining third party innovation [1]. Applications, as add-on software subsystems that connect to digital platforms, are considered to be modules, and the creation and commercial realization of these modules can be considered to be innovation [1, 14].

Some owners curate their digital platforms in order to control for the quality of complements distributed to end users. This occurs as part of the implementation

1530-1605/16 $31.00 © 2016 IEEE
DOI 10.1109/HICSS.2016.163

phase of a process of innovation [17], just before developers' code is potentially commercialized. In this way the process of managing and deciding upon the innovation of digital platform complements is a process of IS control. For example, in the case of Apple's iOS platform, complements in the form of apps are formally curated in its App Approval Process, where code submitted by third party developers is inspected and judged for admission into the App Store. Following on from this, an aspect of control over the innovation of complements, curation is of interest as negative outcomes can lead to a tension of control and generativity [2], which can erupt into conflict between platform owner and developer.

The notion of a tension between control and generativity in platform innovation is increasingly recognized in IS research [1, 2, 14]. Empirical research also shows that platform owner control over innovation can be challenged by developers [2]. However, a description and explanation of the dynamics concerning the control of innovation of complements on digital platforms has yet to be conducted. In this way an understanding of how a platform owner manages these dynamics may provide an insight into the strategies employed by a platform owner to control the innovation of platform complements.

The objective of this paper is to build on research [18] that develops a type 2 theory [19] which describes and explains our understanding of the dynamics of the control of innovation of platform complements. This research inductively builds a low level process theory that describes and explains the unfolding of a set of actions between platform owner and developer that may occur when code is rejected as a platform complement.

The overall research question addressed in this research is:

How can the dynamics of the control of digital platform innovation be described and explained?

This paper takes this earlier research as a backdrop and reports on one of its key findings. It identifies the strategies a platform owner uses to manage the dynamics of the innovation of platform complements. In doing so it shows how IS can bring a different perspective on traditional product management notions [20] concerning architectural innovation within platforms [5].

2. Platform Innovation and Platforms as Modular Architectures

Modular architectures are composed of loosely coupled components connected with standardized interfaces, such that components can be replaced or modified internally without compromising overall system functionality [21]. They are governed by a set of design rules which define how the functionality of an architecture is partitioned into modules [22]. Design rules specify what modules are contained within a system, what their functionality is, and how these modules interface and interact. These aspects of the design rules are made visible to third parties who can potentially contribute modular functionality to a system's architecture. Other aspects of design rules remain invisible to these parties. These typically concern the internal arrangement of functionality within modules, which are black boxed and hidden to the rest of the systems architecture.

An understanding of the modular composition of architecture provides insights into the specificities of system innovation. Modular system architectures allow for types of innovation beyond the dichotomy of incremental or radical innovation [20]. These systems are open to modular innovation for the creation of new modules or innovation within existing modules, as well as architectural innovation which concerns new arrangements of design rules between system modules. The implications of modularity on innovation extend beyond this however. Loose coupling between modules allows for the disintegration of vertical and horizontal organizational structures, which enables modules of a system to be worked upon by a decentralized production network [21]. This facilitates a greater rate of innovation as organizations, unencumbered by dependencies, can work on different modules of the same system in parallel.

Platforms embody modular architectures composed of two sets of modular components. The first, the platform, is a core module, owned by a platform owner, and offers up functionality through interfaces of low variability to the second, which are complementary modular components of high variability and built by third parties [5]. The traditional literature concerning platform innovation assumes that the division of activities and responsibilities between actors involved in innovation is both stable and controllable [3]. Platform design rules represent a stable set of long lived standardized interfaces upon which peripheral complementary modules rely for drawing functionality from the core platform module [5].

In terms of increasing the rate of innovation, the opening up of platform architectures to third parties provides opportunities for the creation of many new and novel platform complements [6]. However, whilst complements are free to change over time, the stability of the design rules indicates that platform architecture evolves slowly and that architectural innovation is a relatively rare event [5].

Furthermore, the consensus in the platform literature [23] is that platform owners control changes to architecture: platform owners have "decision rights that determine who can interact with or modify which components in what ways" [5, p25]. The platform owner is therefore the ultimate authority with regards to design rules [22] which guide decisions such as whether new types of platform complement should be permitted.

It becomes possible to see how these assumptions stand up in the context of digital platforms by studying instances of contested innovation in curated platforms.

3. Research Methodology

An approach is needed to study the series of actions that occur between a platform owner and developers during instances of contested innovation of platform complements, so that a process theory can be built inductively. This theory is required to identify and describe the underlying structural properties of the patterns of actions across examples of contested innovation. The theory is also required to explain mechanisms that lead to the process unfolding as it does. In doing so this research goes some way to address Pentland's [24] call for studies in IS that treat patterns of action as objects of enquiry.

This research uses the approach of narrative networks, defined as "a collection of functional events related by their sequential occurrence in a story or set of stories" [25, p244], to describe sequential patterns of action as graphs. In order to advance a story plot, each functional event is represented as a narrative fragment which contains two or more actants and an action that links them [26]. Graphically a narrative network consists of nodes linked together by ties. Each node represents a narrative fragment, and each link, tying one narrative fragment to the next, represents sequential progression from one fragment to the next, which in turn advances the plot. Not only do narrative networks summaries individual sequences of action, they enable many sequences to be plotted on the same graph. This allows for the comparison of sequences and for pattern matching, as well as a foundation for explaining the unfolding of patterns of actions.

This research uses Apple's iOS platform ecosystem as a source for examples of contested innovation of platform complements, in the form of apps, on a curated digital platform. It focusses purely on sequences of actions that unfold between Apple, as platform owner, and developers. Qualitative data concerning the actions of developers attempting to progress the release of their code as apps on Apple's App Store and Apple attempts at controlling this were sourced from blog posts. These blog posts were sourced from "Tech Blogs" [27] which comment on technological innovation such as mobile technology and apps, and on the high tech companies, including Apple, Google and Microsoft, that facilitate this. Data was collected from blogs posted in the three year period from the start of 2007 until the end of 2011. In total data was extracted from 4664 blog entries specifically reporting on tensions between developers and Apple.

A grounded approach [28] was taken to analyzing the data and to constructing narrative networks based around common sets of actions. The decision to use a grounded approach was taken for two reasons. First, there is an absence of theory to describe the phenomenon in question, which provides an opportunity for theory to be developed inductively [28]. Second, the phenomenon that is studied here is inherently processual, which given its sequential and changeable nature, makes it suited to a grounded approach to theory building.

Open coding [29] was conducted on the corpus of data. In conducting the open coding, narrative fragments, or "codable moments", related to the phenomenon under study [30] and linked to specific apps were identified. These fragments describe instances of developers' actions attempting to implement a specific app, Apple's action to control this implementation, developers' follow-up responses to Apple's control actions, and so on.

Once the corpus of data was thoroughly parsed and an exhaustive list of narrative fragments were identified, they were then clustered around individual apps and sequenced into stories concerning actual instances of contested innovation. Table 1 lists the 45 stories of contested apps found in the data. Table 2 then illustrates a sequence of narrative fragments describing the story of 3G Skype.

The narrative fragments were then reanalyzed to develop a set of axial codes [29] across the contested apps that identify actions describing the respective moves by Apple and developers. This allowed the narrative fragments around each contested app to be abstracted and recast as a sequence of actions, which in turn allows comparison of sequences of actions across different contested apps. Through repeated open coding and comparison within and across contested apps, an exhaustive list of possible actions by Apple and developers was inductively generated.

Selective coding [29] was then used to group these initial actions into generic categories of actions. From this, four generic actions by developers (requesting, bypassing, influencing, and revising) and another four generic actions by Apple (allowing, blocking, refining, and ignoring) were identified. The coding scheme that was generated to describe the observed sequences of actions is shown in table 3.

Box Office	C64 Emulator	Hottest Girls	Stanza	Trillian
3G Skype	CastCatcher	I Am Rich	NinjaWords	Tweetie
Admob	Convertbot	Jailbreakme	Readability	VoiceCentral
Adobe Flash	Google Books	Mark Fiore	Routesy	Tawkon
Adobe dev. tools	Eucalyptus	Me So Holy	Sekai Camera	Wallpaper Universe
Ari David	EyeTV	Nescaline	Simply Beach	Wi-Fi Sync
Baby Shaker	Fin. Times	Netshare	Someecards	GV Mobile
Big Brother Security	Wallpaper Gallery	EFF Updates App	Pulse News Reader	Pull My Finger
Bobble Rep	Google Voice	Opera	Podcaster	TrapCall

Table 1. List of 45 stories concerning contested innovation of complements.

Apple and Skype tussle over whether to enable voice over IP calls on 3G, rather than just Wi-Fi, using Skype's iPhone app	
Fragment#1	Skype wish to launch a full 3G version of their app on the iPhone including 3G access
Fragment#2	Apple prevents a full 3G variant on its App Store, due to AT&T demands to prevent the 3G variant.
Fragment#3	Skype publically endorses groups lobbying the FCC to force VoIP apps like Skype to be allowed on AT&T's 3G
Fragment#4	Apple and AT&T eventually give in to allow VoIP apps like Skype over 3G

Table 2. Narrative fragments concerning 3G Skype

Category	Codes	Explanation
Progressing developer's implementation of code	Request	Developer requests Apple to accept code as a valid App and to distribute it from Apple App Store
	Bypass	Developer bypasses Apple and its App Store to distribute code as an App through some other source. (e.g. from an alternative App Store, or as a Web App directly from the internet)
	Influence	Developer attempts to persuade Apple directly or indirectly to allow code to be implemented as an App and distributed from Apple App Store
	Regroup	Developer abandons attempts at implementing code in its current state as an App, and adjusts code to be acceptable to Apple, starts all over again, or simply gives up.
Progressing Apple's control of innovation	Allow	Apple allows developer code to be implemented as an App and distributed from the Apple App Store
	Block	Apple explicitly refuses developer code to be implemented as an App and distributed from the Apple App Store
	Refine	Apple changes or reinterprets its rules controlling the type of content that is admitted into the App Store, thus allowing certain instances of code to be implemented as Apps.
	Ignore	Apple ignores the actions of the developer, which is in effect an implicit blocking action

Table 3. Codes describing the actions of Apple and the developers

The narrative fragments making up the 45 established stories were then coded using the scheme, until 45 sequences of coded actions were obtained. Table 4 illustrates the coded sequence of actions for the 3G Skype story. Appendix A shows all 45 stories coded as generic sequences of action.

	3G Skype
Action#1	Developer **Requests** Platform Owner
Action#2	Platform Owner **Blocks** Developer
Action#3	Developer **Influences** Platform Owner
Action#4	Platform Owner **Refines** Developer

Table 4. The story of 3G Skype as a sequence of coded actions.

Each of the 45 coded stories was then represented graphically as a narrative network, as shown in figure 1.

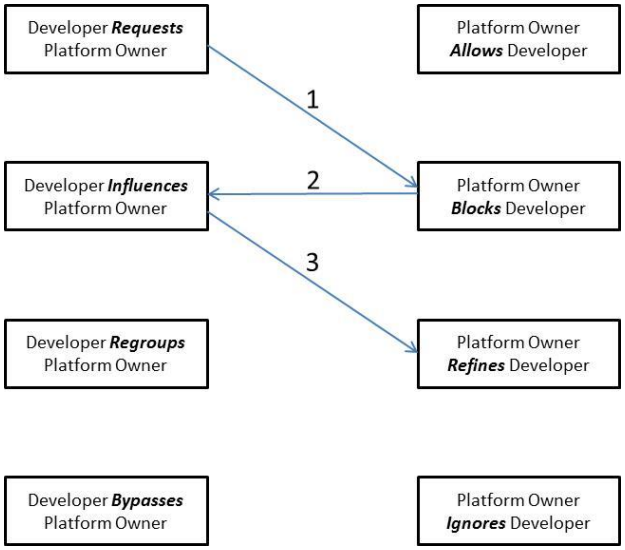

Figure 1. The story of 3G Skype as a narrative network.

Finally, the 45 individual narrative networks were then combined and normalised into a single "normalised" narrative network for analysis, as represented in figure 2. Here the thickness of the ties is proportional to the number of instances of transiting from one node to another.

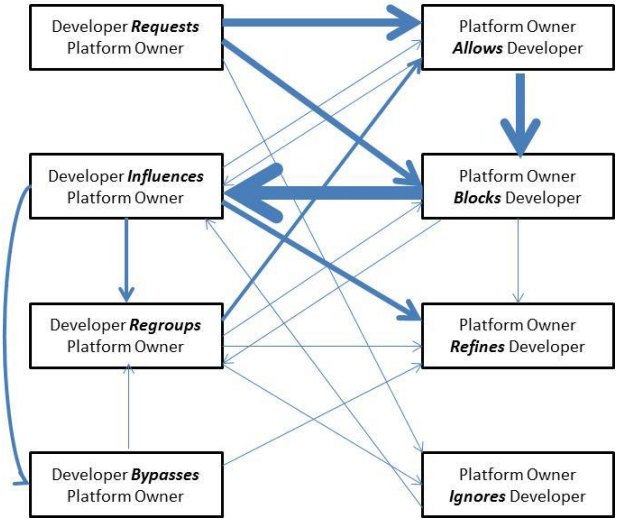

Figure 2. 45 Coded stories combined as a normalised narrative network.

In addition, two further sources of data were used to explain the mechanisms that Apple employs to control platform innovation. The first concerns the iOS Developer Program License Agreement that Apple

puts in place between itself and third party developers. The second concerns the App Store Review Guidelines. Both documents detail obligations and responsibilities on developers and rules concerning the development of iOS apps.

4. Analysis

A summary of the research analysis is now given, focusing first on describing the dynamics of the contested innovation of complements, before moving on to explaining these dynamics.

4.1. Description

By abstracting the combined narrative network and focusing on the most frequently occurring sequences it was possible to identify a generalized sequence of actions that was broadly common across the stories (figure 3a).

This generalized sequence of actions was split into three distinct stages, which describe the dynamics of the control of innovation of platform complements. The first stage (figure 3b) represents a build-up to tension between Apple and a developer and concludes in Apple just blocking (or allowing, changing its mind, and then blocking) a developer's request for an innovation. The second stage (figure 3c) represents a tension between the two parties and describes the developer attempting to influence Apple to reverse its block. The third and final stage (figure 3d) represents a resolution to the tension, and identifies three possible outcomes: Outcome 1) Apple responds positively to attempts at influencing it and allow the complement onto its platform, possibly refining its rules at the same time; Outcome 2) The developer bypasses Apple's App Store, Platform Ecosystem and control and distributes its unchanged complement to iOS devices via other channels; Outcome 3) The developer complies with Apple's reasons for blocking the innovation, and regroups (Outcome 3a) and reworks the complement until it is compliant with Apple's specifications until it is allowed into the App Store (Outcome 3b).

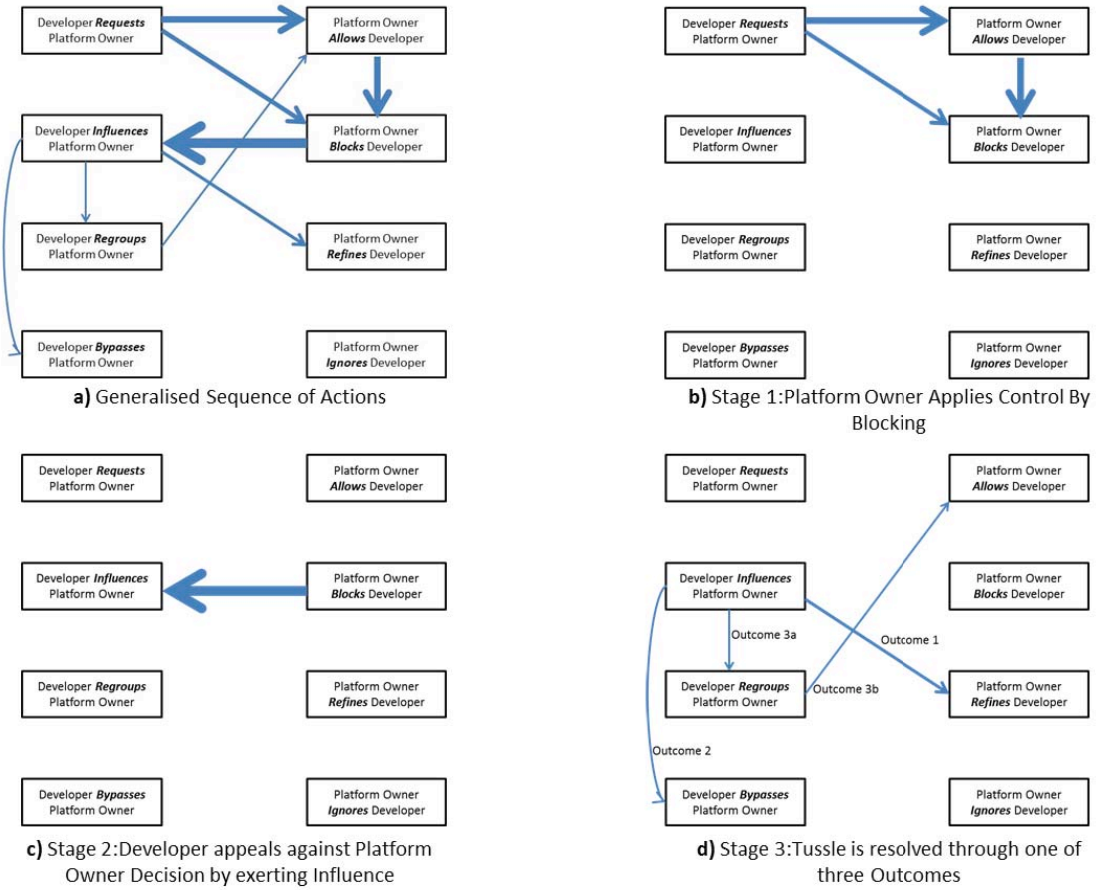

Figure 3. Generalised sequence of actions unfolding in three stages in contested innovation

4.2. Explanation

The observed sequences of actions can be explained by two opposing sets of mechanisms. The first concerns Apple's App Approval Process. The second concerns the developers' three strategies of regrouping, bypassing and influencing. Each is now considered in turn.

Apple's Apps Approval Process acts as a mechanism by which it attempts to control for appropriate complements on its platform. Official Apple documentation, such as the iOS Developer Program License and the App Store Review Guidelines, as well as theoretical concepts of formal organizational control [31-33] provide a means of analyzing this control mechanism. The iOS Developer Program License and the App Store Review Guidelines consist of terms which inform developers by specifying what new apps will or will not be tolerated. It is against these specifications that Apple measures and evaluates developers' code in order to decide whether it should be admitted into the App Store as an innovation. If the code falls within the specifications then the developer is rewarded by having the code admitted as an App into the App Store. If the code falls outside of the specifications then it is rejected. On occasion, Apple will first allow an app and then block, or pull, the same app from its App Store. In these cases Apple either makes an initial mistake in its assessment of an app against its specifications, or changes or reinterprets its specifications at a later date, which then results in an approved app being pulled from the App Store.

Following Apple's block of an app to the App Store, it was found that developers use one or a combination of three mechanisms triggered to counter this controlling action. The developer will first attempt to influence Apple to persuade them to reverse their decision. In the empirical study this mechanism is triggered first by default, simply because the act of being reported in a blog can be thought of as an attempt to influence Apple. If this strategy is unsuccessful the developer may then try one of two actions. First, the developer may continue the struggle and bypass Apple, by distributing their code to iOS users through some alternative channel beyond Apple's control. For example by distributing the code as an HTML 5 "web" app or by distributing the code via an unsanctioned app store open to users who have hacked (jailbroken) their iPhones. Apple's response to such actions is frequently to blackball the developer and to stick to their blocking decision. Alternatively, the developer may capitulate to Apple's control and adjust the blocked app's code until it conforms to Apple's specifications in the hope that Apple will allow it the second time around. In each case, bypassing or regrouping, the original app does not enter the App Store. If, however, an attempt at influencing is successful, then Apple will reverse its block and the original app will make its way into the App Store.

In reversing its block on a new app, it was found that Apple adjusts its control of innovation in one of two ways. It may reverse a block by adjusting the way that it interprets its specifications, whilst keeping the wording of the specification in the appropriate document unchanged. Alternatively, it may change the specification in and of itself to accommodate a new app. Table 5 summarizes the key moves made by Apple, developers and the outcomes for each of the 45 instances of contested innovation that were studied.

Table 5 Column B shows the broad specifications that Apple applies to attempt control. Column C indicates the dominant strategy applied by the developer in each case. Column D shows whether the app eventually ended up in the app store or not. Finally, Column E indicates what approach Apple took towards its specifications as a result of having its decisions contested. Of the 45 instances of contested innovation, 31 eventually were eventually admitted into the App Store. And of these 31 instances, 21 involved some change in the way that Apple controls its App Approval Process, of which 15 involved Apple reinterpreting the specification of its rules concerning what types of new apps will be considered as permissible innovation, and 6 involved it actually changing the specification of its rules.

5. Discussion and Conclusion

The theory developed in the previous section describes a process of change where platform owners assert control by blocking the innovation of undesirable complements, developers attempt to resist these attempts at control, and ultimately some outcome is arrived at in terms of whether innovation is realized or not. Van de Ven et al. [34] identify four broad categories concerning processes of organizational change. These concern processes of evolution, dialectics, teleology and life cycle respectively. The process of change that is theorized in this study is one of dialectics between two sides concerning thesis, antithesis and synthesis. This dialectical process presented here is explained by two opposing sets of mechanisms. On the part of the platform owner there is a process whereby complements are approved for distribution. On the part of the developer there are mechanisms for resistance to control as well as for capitulation. This study is not unique in describing the process of digital innovation as a dialectic.

(Column A) Narrative, number & name	(Column B) Specification of Rule applied by Apple	(Column C) Developer response	(Column D) Final outcome	(Column E) Change in specification
20 Hottest Girls	Objectionable Content	Influence	Not In App Store (14 Apps)	No Change (24 Apps)
21 I Am Rich	Dubious Value			
26 Netshare	Excessive Cellular Data Usage			
44 Wallpaper Universe	Objectionable Content			
6 Baby Shaker	Objectionable Content			
22 Jailbreakme	Jailbreaking	Regroup		
7 Big Brother Security	Data Collection and Privacy Abuse			
16 Financial Times	Alternative Payment Platforms	Bypass		
24 Me So Holy	Objectionable Content			
25 Nescaline	Platform Emulators			
32 Readability	Alternative Payment Platforms			
38 Tawkon	Use of Private APIs			
42 VoiceCentral	Duplication of Functionality			
45 Wi-Fi Sync	Use of Private APIs			
28 Opera	Duplication of Functionality	Regroup	In App Store (31 Apps)	
11 CastCatcher	Excessive Cellular Data Usage			
12 Convertbot	Inappropriate Look and Feel			
17 Google Books	Alternative Payment Platforms			
29 Podcaster	Duplication of Functionality			
31 Pulse Reader	Threat to Apple's Legal Position			
34 Sekai Camera	Use of Private APIs			
36 Someecards	Objectionable Content			
37 Stanza	Use of Private APIs			
3 Adobe Flash	Executable Code from other Platforms	Bypass		Reinterpretation of specification (15 Apps)
18 Google Voice	Duplication of Functionality			
19 GV Mobile	Duplication of Functionality			
13 EFF Updates	Objectionable Content	Influence		
14 Eucalyptus	Objectionable Content			
15 EyeTV	Excessive Cellular Data Usage			
30 Pull My Finger	Objectionable Content			
33 Routesy	Threat to Apple's Legal Position			
35 Simply Beach	Objectionable Content			
39 TrapCall	Unknown			
40 Trillian	Unknown			
41 Tweetie	Objectionable Content			
43 Wallpaper Gallery	Objectionable Content			
5 Ari David	Objectionable Content			
8 Bobble Rep	Objectionable Content			
9 Box Office	Unknown			
1 3G Skype	Enabling VoIP over 3G			Specification Changed (6 Apps)
10 C64 Emulator	Platform Emulators			
2 Admob	Alternative Advertising Platforms			
23 Mark Fiore	Objectionable Content			
27 NinjaWords	Objectionable Content			
4 Adobe developer tools	Cross-Compilers			

Table 5. Summary of outcomes by platform owner control and developer response.

For example Ghazawneh et al [1] describe a dialectical approach to theorizing tensions between control and generativity occurring within processes of digital platform innovation. In this way the theory presented here empirically confirms previous dialectical notions of platform innovation.

Moreover, the dialectical basis of the theory presented can be used to challenge and further contribute to the platform innovation literature. The focus has been thus far on the innovation of platform complements, which taking the perspective of Henderson et al. [20] is incremental or, at best, modular innovation. Apps that are eventually accepted in their original form by Apple following this dialectic represent instances of modular innovation, as they represent a new type of module that was not previously allowed. This occurred on 21 occasions.

However, it would appear that on occasion other forms of innovation are realized through this dialectic. The outcome of some instances of contested innovation is that Apple actually changes its specifications for control. These specifications determine which combinations of modules and interfaces are allowable or not to render functionality in apps. A novel change in the specifications of interfaces is a change in architecture [22], which in turn is an architectural innovation [20]. In this way this dialectic can yield both modular and architectural innovation within platforms. In this way, it would appear that both Apple, as platform owner, and third parties are engaging in practices of open innovation not just in terms of digital services as platform complements, but also in terms of the underlying architectural capabilities of the platform.

Furthermore, the dynamics of this architectural innovation would appear to contradict convention. The literature on platform architectures [5] indicates that through the establishment and ownership of design rules [22] the platform owner has the right to exercise stable control over the innovation of platform architecture. By retaining control over a platform, the platform owner is able to manage the evolutionary trajectory of the platform [35] and regulate the innovation of complements [6]. In addition, the sense in the traditional technology and innovation management literature [20] and the platform architecture literature [5] is that architectural innovation occurs relatively infrequently.

This research provides empirical evidence that questions these notions. Of the 21 occasions when modular innovation occurred, architectural innovation also occurred 6 six times (Table 5 Column E). In these 6 instances Apple's specifications concerning permissible combinations of modules and interfaces were changed. These 6 instances indicate occasions when the actions of developers influenced Apple to make changes to its architecture in ways that was initially unwilling to make. This brings into question the extent to which the platform owner is in control of architectural innovation. In addition, architectural innovation occurred in 6 instances in contrast to 21 occasions of modular innovation, which brings into question the notion that architectural innovation is an infrequent event.

Amongst the capabilities that Apple uses to maintain control over the App Store is its command over its technology. The iPhone operating system is designed such that each device is uniquely constrained to the Apple App Store as a source for new apps. This facilitates Apple's power over developers as they become uniquely dependent on Apple for access to consumers. This allows Apple to set and enforce its terms in the App Approval Process. Apple maintain their monopolistic hold over this single distribution channel through the ownership of critical interface components, such as between the operating system and the App Store, and by maintaining "tight couplings" between these critical architectural components [5] to prevent modification by third parties.

However, the digital characteristics of iOS are such that it can be reprogrammed and modified [15], so that the tight couplings between operating system and official App Store are loosened through a process of hacking or "jailbreaking". As a digital infrastructure [2], the ensemble of the iOS platform, and the components around it, now consist of layered modular architecture [16], which enable the substitution of architectural components beyond Apple's original specifications. In this way iOS consumers are liberated to source applications from alternative app stores outside of Apple's control.

As a consequence the balance of power between platform owner and developer is adjusted. Developers are now free to bypass Apple's App Store and distribute apps to jailbroken iOS devices from alternative app stores. The ability to bypass the App Store is potentially threatening to Apple, as a loss of their installed base of developers and consumers reduces their power to maintain a healthy platform ecosystem [36]. With the threat of being increasingly bypassed and losing an installed base, Apple becomes more open to accommodating developers' demands with regards architectural change in order to maintain its ecosystem. This may account, in part, for more architectural innovation than might otherwise be expected.

This research contains limitations. First, restrictions concerning space have constrained possibilities for providing thick descriptions to further enhance the articulation of the analysis. Second, as a form of

sequence analysis, the use of narrative networks can approach narrative positivism [37], where the focus on events comes at the expense of the richness of context, whilst lacking the statistical rigor of quantitative research. However after the parsing of 4664 blog entries and the reconstruction and analysis of 45 individual stories, the patterned sequences of action that unfold and the mechanisms behind them are convincing.

Future work may focus on addressing the need for more attention to breadth or depth. However, an equally interesting way forward is to investigate how the platform owner's management of this dynamic varies in different types of digital infrastructure. In the context of mobile platforms, apps and operating systems reside on the IT artefact as "native code". In these environments it is easier to access code and to pick apart the tight couplings of control. In the context of other IS, such as the cloud, code reside and executes remotely on virtualized computing assets, secured from users and other parties. In these environments it is harder to access and to pick apart the tight couplings of control and this might change the dynamics of platform innovation again. In the cloud, there is also potential for further control of innovation of complements, as platform development environments can, in the context of platform as a service (PaaS), be run centrally under the auspices of the platform owner, allowing for behavioral as well as outcome control [33]. How the context of these different types of digital infrastructures would alter platform owners' strategies for responding to challenges to its authority regarding the control of the innovation of platform complements would make for interesting further research.

To conclude, this research is of interest for three reasons. First, it treats patterns of actions as objects of enquiry, which is an uncommon approach requiring more attention in IS [24]. Second, it fills a genuine gap in the growing IS literature concerning platform innovation by providing an understanding of the dynamics of digital platform innovation [14] as well as of the strategies for how they are managed. Last, it provides an opportunity to use IS notions of digitalization [38] and layered modular architecture [16] in order to challenge understandings of architectural innovation taken from the product development [20] and technology strategy literature [5]. In this way this research demonstrates that some platform owners and third parties engage in practices of open innovation [12, 13] not just for the production of digital services as platform complements, but also in terms of evolving the underlying architectural capabilities of the platform in and of itself.

6. References

1 Ghazawneh, A., and Henfridsson, O.: 'Balancing platform control and external contribution in third-party development: the boundary resources model', Information Systems Journal, 2012

2 Tilson, D., Sørensen, C., and Lyytinen, K.: 'Change and Control Paradoxes in Mobile Infrastructure Innovation: The Android and iOS Mobile Operating Systems Cases'. Proc. 45th Hawaii International Conference on System Science (HICSS 45), Maui, Hawaii, USA2012 pp. Pages

3 Gawer, A.: 'Platform Dynamics and Strategies: from Products to Service', in Gawer, A. (Ed.): 'Platforms, Markets and Innovation' (Edward Elgar Publishing Limited, 2009), pp. 45-76

4 Authors own paper

5 Baldwin, C.Y., and Woodard, C.J.: 'The Architecture of Platforms: A Unified View', in Gawer, A. (Ed.): 'Platforms, Markets and Innovation' (Edward Elgar, 2009), pp. 19-44

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

7 Iansiti, M., and Levien, R.: 'The keystone advantage: What the new dynamics of business ecosystems mean for strategy, innovation and sustainability Harvard Business School Press', Boston, MA, 2004

8 Rochet, J.C., and Tirole, J.: 'Platform competition in two-sided markets', Journal of the European Economic Association, 2003, 1, (4), pp. 990-1029

9 Katz, M.L., and Shapiro, C.: 'Technology Adoption in the Presence of Network Externalities', Journal of Political Economy, 1986, 94, (4)

10 Hart, O., and Moore, J.: 'Property Rights and the Nature of the Firm', Journal of Political Economy, 1990, pp. 1119-1158

11 Farrell, J., and Katz, M.L.: 'Innovation, rent extraction, and integration in systems markets', The journal of industrial economics, 2000, 48, (4), pp. 413-432

12 Chesbrough, H.W.: 'Open innovation: The new imperative for creating and profiting from technology' (Harvard Business Press, 2003. 2003)

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

14 Tiwana, A., Konsynsky, B., and Bush, A.A.: 'Research Commentary - Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics', Information Systems

Research, 2010, (Articles in Advance - Published online ahead of print Novemeber 18, 2010), pp. 1-13
15 Kallinikos, J., Aaltonen, A., and Marton, A.: 'The Ambivalent Ontology of Digital Artifacts1', Management Information Systems Quarterly, 2013, 37, (2), pp. 357-370
16 Yoo, Y., Henfridsson, O., and Lyytinen, K.: 'The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research', Information Systems Research, 2010, 21, (5 - 20th Anniversary Special Issue of Emerging Challenges)
17 Van de Ven, A.H., Polley, D.E., Garud, R., and Venkatraman, S.: 'The Innovation Journey', in Editor (Ed.)^(Eds.): 'Book The Innovation Journey' (Oxford University Press, 1999, edn.), pp.
18 Authors own research
19 Gregor, S.: 'The nature of theory in information systems', MIS Quarterly, 2006, 30, (3), pp. 611-642
20 Henderson, R.M., and Clark, K.B.: 'Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms', Administrative Science Quaterly, 1990, 35, (1), pp. 9-30
21 Langlois, R.N., and Robertson, P.: 'Networks and Innovation in a Modular System: Lessons from the Microcomputer and Stereo Component Industries', Research Policy, 1992, 21, pp. 297-313
22 Baldwin, C.Y., and Clark, K.B.: 'Design Rules: The Power of Modularity, Volume 1' (MIT Press, 2000. 2000)
23 Eisemann, T.R., Parker, G., and Van Alstyne, M.: 'Opening Platforms: how, when and why?', in Gawer, A. (Ed.): 'Platforms, Markets and Innovation' (Edward Elgar Publishing Limited, 2009), pp. 131-162
24 Pentland, B.T.: 'Desperately seeking structures: grammars of action in information systems research', ACM SIGMIS Database, 2013, 44, (2), pp. 7-18
25 Pentland, B.T., and Feldman, M.S.: 'Designing routines: On the folly of designing artifacts, while hoping for patterns of action', Information and Organization, 2008, 18, (4), pp. 235-250
26 Pentland, B.T., and Feldman, M.S.: 'Narrative networks: Patterns of technology and organization', Organization science, 2007, 18, (5), pp. 781-795
27 Davidson, E., and Vaast, E.: 'Tech talk: An investigation of blogging in technology innovation discourse', Professional Communication, IEEE Transactions on, 2009, 52, (1), pp. 40-60
28 Eisenhardt, K.M.: 'Building theories from case study research', Academy of Management Review, 1989, pp. 532-550
29 Strauss, A., and Corbin, J.: 'Basics of qualitative research: Grounded theory procedures and techniques' (Sage Publications, Inc., 1990. 1990)
30 Boyatzis, R.E.: 'Transforming qualitative information: Thematic analysis and code development' (Sage, 1998. 1998)
31 Ouchi, W.G.: 'The relationship between organizational structure and organizational control', Administrative Science Quarterly, 1977, 22, (1), pp. 95-113
32 Ouchi, W.G.: 'The transmission of control through organizational hierarchy', Academy of Management Journal, 1978, 21, (2), pp. 173-192
33 Ouchi, W.G.: 'A conceptual framework for the design of organizational control mechanisms', Management Science, 1979, 25, (9), pp. 833-848
34 Van de Ven, A.H., and Poole, M.S.: 'Explaining development and change in organizations', The Academy of Management Review, 1995, 20, (3), pp. 510-540
35 Schilling, M.A.: 'Protecting or Diffusing a Technology Platforms: Tradeoffs in Appropriability, Network Externalities, and Architectural Control', in Gawer, A. (Ed.): 'Platforms, Markets and Innovation' (Edward Elgar Publishing, Ltd., 2009), pp. 192-218
36 Gawer, A., and Cusumano, M.A.: 'How Companies Become Platform Leaders', MITSloan Management Review 2008, 49, (2), pp. 28-35
37 Abbott, A.: 'From causes to events', Sociological Methods & Research, 1992, 20, (4), pp. 428
38 Tilson, D., Lyytinen, K., and Sørensen, C.: 'Digital Infrastructures: The Missing IS Research Agenda', Information Systems Research, 2010, 21, (5 - 20th Anniversary Special Issue of Emerging Challenges)