Karhu, Kimmo; Gustafsson, Robin; Eaton, Ben; Henfridsson, Ola; Sørensen, Carsten

Four Tactics for Implementing a Balanced Digital Platform Strategy

Published in:
MIS Quarterly Executive

DOI:
10.17705/2msqe.00027

Published: 01/06/2020

Document Version
Peer reviewed version

Please cite the original version:
Karhu, K., Gustafsson, R., Eaton, B., Henfridsson, O., & Sørensen, C. (2020). Four Tactics for Implementing a Balanced Digital Platform Strategy. *MIS Quarterly Executive*, *19*(2), 105-120. [4].
https://doi.org/10.17705/2msqe.00027

Four Tactics for Implementing a Balanced Digital Platform Strategy

Digital platforms, such as Apple's iOS and Google's Android, face two major challenges in maintaining their competitive positions. First, the ever-increasing variety of third-party apps risks fragmenting a platform and requires the platform owner to use controls to maintain platform unity. Second, a too-open approach may invite competitors to exploit a platform. Thus, a balanced digital platform strategy requires a platform owner to deploy four tactics—leverage, control, exploit, and defense—to make the necessary trade-offs between variety and unity, and open and closed.[1,2]

Kimmo Karhu
City of Helsinki and Aalto University (Finland)

Robin Gustafsson
Aalto University (Finland)

Ben Eaton
Copenhagen Business School (Denmark)

Ola Henfridsson
University of Miami (U.S.)

Carsten Sørensen
The London School of Economics and Political Science (U.K.)

Two Strategic Trade-offs for a Digital Platform

In little more than a decade, Apple's iOS and Google's Android have each managed to attract over two million apps that are built on top of the platforms. Apps and other types of value-adding complements[3] are imperative for digital platforms and help platforms to reach markets that would otherwise be unavailable. Our research is aimed at identifying the strategic challenges that digital platform owners face as their platforms expand and how they can cope with these challenges.

The need to understand these challenges is increasing as platforms expand beyond smartphones and tablets to smartwatches and other embedded devices. Home-automation, self-driving cars, network-enabled trash cans, drone swarms, digital twins and other technologies commonly known as the Internet of things (IoT) will create a diversity of needs to transform business models based on digital platforms. It is likely

---

[1] Jens Dibbern is the accepting senior editor for this article.
[2] The authors are thankful for the helpful guidance and comments by Professor Dibbern and the anonymous reviewers throughout the review process.
[3] Complements are value-adding "goods" built on top of a digital platform by third-party producers (complementors). Examples of complements include smartphone apps, applications built on top of the salesforce.com platform and business information added to a mapping platform such as Google Maps. Complementarity means that the value of a complement and the platform together is more than separately.

that this digital transformation[4] will not only be captured by FAANG (Facebook, Apple, Amazon, Netflix, Google) and BAT (Baidu, Alibaba, Tencent) companies, but will also represent strategic challenges and opportunities for a range of traditional firms not previously involved in building digital platforms.

Traditional firms and new platform entrants can learn from two fundamental strategic trade-offs that the front-runners have had to make. The first trade-off—*variety vs. unity*—is between attracting a variety of complements while simultaneously preserving the platform's unity.[5] Although digital technologies provide almost unbounded opportunities for innovation, this characteristic can result in low-quality or even harmful complements. A key challenge for platform owners is that the large number of third-party complement providers (i.e., complementors) is far beyond what can be controlled on a one-on-one basis.

The second trade-off—*open vs. closed*—concerns the need to balance the openness of a platform to attract collaborators while at the same time avoiding exploitation by competing platforms. If a platform is too open, competing platforms may then exploit the platform's open and shared resources. It is much easier to replicate, reengineer, and exploit digital resources than physical ones. Consider the case of Google, which established the Android Open Source Project (AOSP) to attract more device manufacturers to the Android platform. However, Amazon exploited the openness of Android to build its competing Android-like platform, Amazon Fire. As a consequence, Amazon it is now the world's third-largest tablet vendor. While sharing the same base technology, the Google and Amazon platforms compete against each other. (The strategy followed by Amazon to create Amazon Fire from Google's Android is known as "platform forking."[6]) Amazon monetizes Amazon Fire content and apps in a way that provides no revenue or benefit to Google.

This article shows how incumbent digital platform firms, as well as traditional firms wanting to become platform players or platform complementors, can balance these two trade-offs. Based on our previous research in which we investigated seven digital platforms (see the Appendix for details), we found that four digital platform tactics—*leverage, control, exploit* and *defense*—are needed to balance these strategic trade-offs.

---

[4] For a synthesis of successful digital transformation cases, see: Sebastian, I. M., Ross, J. W., Beath, C., Mocker, M., Moloney, K. G., and Fonstad, N. O. 2017. "How Big Old Companies Navigate Digital Transformation," MIS Quarterly Executive, pp. 197–213.

[5] For more on change and control paradoxes, see: Tilson, D., Sørensen, C. and Lyytinen, K. "Change and Control Paradoxes in Mobile Infrastructure Innovation: The Android and iOS Mobile Operating Systems Cases," *Proceedings of the Annual Hawaii International Conference on System Science, January* 2012, pp. 1324-1333.

[6] For an in-depth study on exploitative platform forking strategies involving the Android platform, see Karhu, K., Gustafsson, R. and Lyytinen, K. "Exploiting and Defending Open Digital Platforms with Boundary Resources: Android's Five Platform Forks," *Information Systems Research* (29:2), May 2018, pp. 479-497, available at https://doi.org/10.1287/isre.2018.0786.

We also found that “boundary resources”[7] are crucial elements in executing the tactics. Boundary resources are the interfaces, tools and rules used to enable, facilitate and control an arm’s-length relationship between a platform owner and third-party complementors.

For the variety vs. unity trade-off, boundary resources should not only be viewed as the tools to enable innovation but also as the means to control third-party complementors and ensure they produce high-quality complements. For the open vs. closed trade-off, boundary resources are not only the tools for defining platform openness but also for defending against and curtailing competing platforms’ possible exploit tactics.

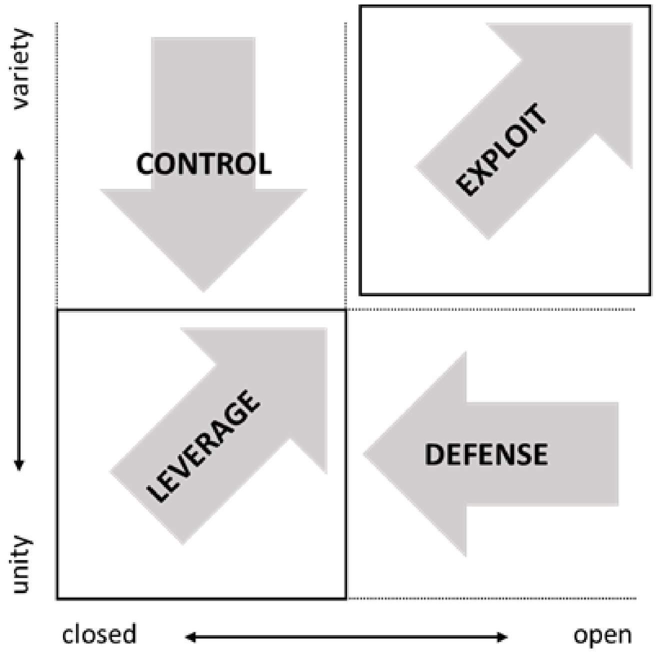
Below, we illustrate the use of the four digital platform tactics by three selected case platforms—Apple iOS, Google Android and Amazon Fire. We conclude the article with four recommendations that provide specific advice for three types of platform participant—platform owner, platform complementor and new platform entrant.

Deploying the Four Tactics for Balancing the Two Digital Platform Trade-offs

To balance the variety vs. unity and open vs. closed trade-offs, platform owners need to master the four types of digital platform tactics—leverage, control, exploit and defense. As depicted in Figure 1, these tactics occupy different spaces defined by the variety/unity and open/closed dimensions. Leverage tactics (bottom left) are deployed to grow the platform in terms of its user and complementor bases. Control tactics are deployed to balance concerns about variability originating from complements with keeping the platform united and consistent (i.e., to address the variety vs. unity trade-off). Note that strong leverage tactics together with too much openness can expose a platform to competitors that may attempt to use exploit tactics to create a competing new and separate platform (depicted as the detached top-right box in Figure 1). Defense tactics are used to address the open vs. closed trade-off by striking a balance between having an open platform that attracts collaborators and one that seeks to prevent competitors exploiting the platform.

[7] For a description of the boundary resources concept, see Ghazawneh, A. and Henfridsson, O. “Balancing platform control and external contribution in third-party development: The boundary resources model,” *Information Systems Journal* (23:2), March 2013, pp. 173-192.

Figure 1: Four Tactics for Balancing the Digital Platform Trade-offs



Boundary Resources Are Crucial for Executing Digital Platform Tactics

Boundary resources are a platform owner's key tools for executing digital platform tactics. Table 1 lists typical uses of boundary resources for each of the four tactics. With the *leverage* tactic, interfaces—for example, application programming interfaces (APIs)—open up the platform to third-party complementors, and tools such as software development kits (SDKs) help developers to innovate and provide new apps. With the *control* tactic, platform rules (typically in the form of software licenses and guidelines) ensure that complementors innovate within predetermined constraints.

A diverse set of complements will make a platform more attractive. However, the multiple complementors involved may cause the platform to fragment (i.e., become less unified), especially when these third parties circumvent or even challenge the platform's existing boundary resources (e.g., by extending the platform or contravening its established design principles).[8] This may lead to competing platforms taking advantage of too-loose boundary resources (e.g., open-source licenses and other terms and conditions), and as a result adopt the *exploit* tactic to use resources from the host platform. To counter this threat, the host platform can use *defense* tactics such as closing up open-sourced code or speeding up API development by using a client-library boundary resource.

---

[8] For a description of distributed tuning of boundary resources, see Eaton, B. D., Elaluf-Calderwood, S. Sørensen, C. and Yoo, Y. "Distributed Tuning of Boundary Resources: The Case of Apple's iOS Service System," *MIS Quarterly Special Issue: Service Innovation in a Digital Age* (39:1), March 2015, pp. 217-243.

Table 1: Typical Uses of Boundary Resources for Each Tactic

Tactic	Boundary Resource	Typical Use
Leverage	API	An application programming interface opens up platform functionality or data for developers.
	SDK	Software development kits are bundles of development tools and APIs to make it as easy as possible for developers to re-use and build upon the platform.
	HAL, CDD CTS	Hardware-abstraction layer, compatibility definition document and compatibility test suite (in Android) are boundary resources for device manufacturers similar to APIs and SDKs for developers.
	Open-source license	In addition to only offering APIs, the platform itself (i.e., the implementing code behind APIs) can be open by using an open-source license. Similar to software, content can also be open by using open content licenses, such as Creative Commons licenses (e.g., Wikipedia).
Control	Publisher terms and conditions	The terms and conditions (e.g., monetization, distribution and banned content) for publishing complements. For example, Apple restricts iOS apps to publishing exclusively in the official Apple App Store.
	Guidelines	Guidelines help to produce higher-quality, unified and compatible complements.
	Client library	A client library wraps platform APIs into an updateable package that helps the platform owner control device fragmentation by updating developer APIs independently of device manufacturers' releases (e.g., Google Play Services in Android).
	MADA	A mobile application distribution agreement defines the rules for Android device manufacturers for how Google Services should be placed in the device.
Exploit	Open-source licenses	If platform resources are published under an open-source license, then exploiters can simply copy them to avoid upfront costs in creating their own modified platform. Permissive open-source licenses (e.g., Apache used in Android) only require attribution, making it particularly easy to exploit.
	Publisher terms and conditions	If a platform publisher's terms and conditions allow alternative app stores (as in Android), an exploiter can also easily exploit the platform's apps by allowing developers to multi-home to its own store.
Defense	Open-source licenses	Reciprocal open-source licenses (e.g., GPL) force license users to share and license changes under the same terms and thus can help to defend the platform by also forcing exploiters to open source their modified platform. Alternatively, a platform can close up open-sourced code.
	Client library	By speeding up API releases, a client library makes it more difficult for API copycats to keep up with platform development.

A Complete Digital Platform Strategy Requires a Combination of Tactics

A complete digital platform strategy requires a combination of the four tactics. Depending on a platform's business context, the four tactics can be employed in various combinations and to differing degrees. Table 2 shows the combinations used by each of our three case platforms and their strengths (i.e., how strongly a platform has used a particular tactic in its strategy). Note that, rather than representing a particular point in time, the table provides an aggregated estimate of the extent to which the platforms used the various tactics over the period 2007-2018.

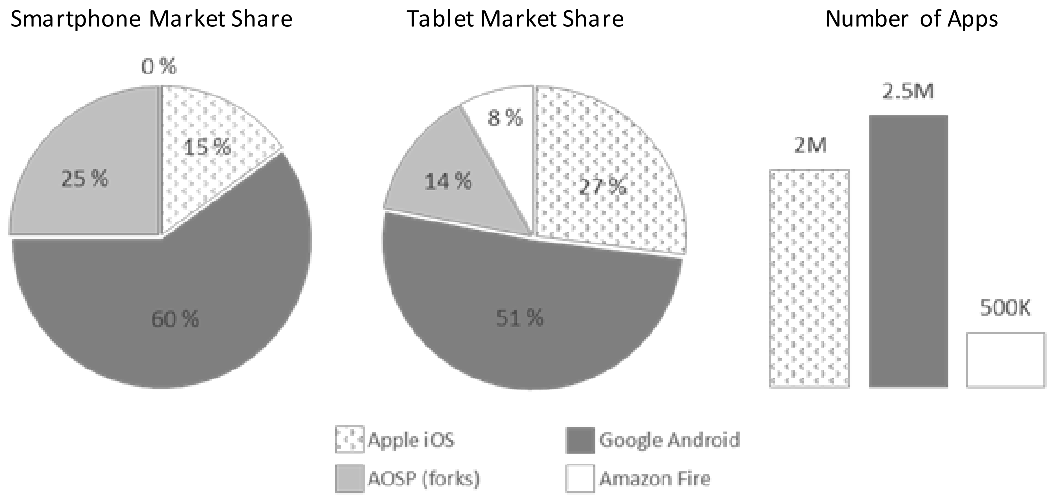
Table 2: Combinations of Tactics Used by the Case Platforms (Aggregated Over 2007-2018)

Case Platform	Leverage Tactic	Control Tactic	Exploit Tactic	Defense Tactic
Apple iOS	++	+++	not used	+
Google Android	+++	++	++	++
Amazon Fire	+	+	+++	not used
+ Used to limited extent; ++ Used to a moderate extent; +++ Used to great extent				

Of the three, Apple led the way in strongly controlling the quality of its iOS platform as it targeted the high-end of the smartphone market. In response, Google strongly leveraged its approach by adopting an open Android platform (which involved less control) and also used exploit tactics to challenge Apple's dominant position. As the latecomer, Amazon with its Fire platform had to focus on exploit tactics; it placed less emphasis on leverage and control tactics because it targeted only the smaller tablet and associated content markets. By adopting an open strategy for Android, Google had to deploy defense tactics against platform forks such as Amazon Fire.

The different platform strategies adopted by Apple, Google and Amazon have resulted in different market outcomes (see Figure 2). The strong focus on leverage tactics by Google (for Android) and Apple (for iOS) has led to them capturing 75% or more of the smartphone and tablet markets between them. Both now have more than two million apps in their app stores.

Figure 2: Smartphone and Tablet Market Shares of Case Platforms, 2018



Source: Estimates from idc.com and statista.com

Figure 2 also shows (in light gray) the market shares of devices based on Android platform forks created via the AOSP. These devices account for roughly one-third of Android-based devices and include all Android devices sold in China.[9] They are not included in the Android market share in Figure 2 because they bring virtually no business to Google.

The Amazon Fire platform, which is also an AOSP fork but in Western markets, falls far behind the Android and iOS platforms with a less than 10% of tablet market share. Instead of actively using leverage and control tactics, Amazon has a more passive approach; it sells the devices at close to or under manufacturing cost as a channel for its digital content.

Overall, Google's Android strategy illustrates that making strong use of all four digital platform tactics leads to success in the digital platform business.

We now describe how our three case platforms used the four tactics to balance the two digital platform trade-offs.

---

[9] For a discussion on the installed bases of "full" Android devices and "forked" Android devices, see Ahonen, T. T. *We Can Now Estimate Global Android Forked Installed Base ie AOSP Devices vs 'full Google' Android,* available at https://communities-dominate.blogs.com/brands/2017/05/we-can-now-estimate-global-android-forked-installed-base-ie-aosp-devices-vs-full-google-android.html.

How the Case Companies Balanced the Digital Platform Trade-offs

We now describe how Apple, Google, and Amazon used the four tactics to balance the two digital platform trade-offs. Additional information about the development of these three case platforms can be found in the Appendix.

Apple Balanced the Variety Vs. Unity Trade-off in Favor of Unity for the iOS Platform

The most important leverage tactic in smartphone platform strategies has been to promote platform growth by generating cross-side network effects[10] between developers and users. A positive network effect arises from the virtuous circle of growth that occurs when growth on one side of the platform (e.g., the developer side) attracts growth on the other (e.g., the user side), and vice versa.

Apple's launch of its App Store is the quintessential example of the successful use of this leverage tactic. The tactic was executed by combining the use of several boundary resources, in particular a software development kit (SDK) and application programming interfaces (APIs). The SDK provides an easy way for developers to build new apps through the use of high-level software constructs in a way that forces them to follow the terms and conditions and thus ensure the quality of their outputs. APIs open up and make platform functionality available for developers to create applications. The App Store is the market intermediary, or matchmaker, that connects users to relevant developers and their apps. The store also provides a monetization mechanism for developers to generate revenue.

While leverage tactics provide a means to encourage innovation by platform complementors, their efforts can lead to undesirable complements. To address this issue, control tactics are used to counter leverage tactics to ensure that complements benefit the platform owner and members of the wider ecosystem. Apple is known for its strict control of the iOS platform,[11] and uses combinations of boundary resources to control the type of apps that it admits into its App Store. Apple's app approval process scrutinizes third-party apps submitted by developers to ensure that their form and function complies with policies set out in its app-approval guidelines and in its publisher terms and conditions, as specified in developer license agreements. An app is rejected if it is found to contravene these strict policies. However, developers have challenged Apple's decisions to exclude apps and this has resulted in the company adjusting its policies on several occasions.

---

[10] For a primer on two-sided markets and network effects, see Eisenmann, T., Parker, G. and Van Alstyne, M. W. "Strategies for Two-Sided Markets," *Harvard Business Review* (84:10), October 2006. p. 92-101.

[11] For in-depth studies on how Apple used boundary resources to control iOS app developments, see: Ghazawneh, A. and Henfridsson, O., op. cit., March 2013; and Eaton, B. D., Elaluf-Calderwood, S. Sørensen, C. and Yoo, Y., op. cit., March 2015.

Google Balanced the Variety Vs. Unity Trade-off in Favor of Variety for the Android Platform

Leverage tactics can also be used to balance the variety vs. unity trade-off in favor of variety instead of controlling for unity and quality. Google came to the market after Apple and, to challenge Apple's dominance, had to adopt additional leverage tactics in its Android platform strategy. Google has built a "multisided" platform[12] that, in addition to users and app developers, has also been opened up to device manufacturers to leverage further network effects.

Google needed to orchestrate several boundary resources to leverage an ecosystem of device manufacturers. First, it defined rules within an alliance agreement to manage Android device manufacturers, such as Samsung, which is part of its Open Handset Alliance (OHA).[13] Second, it has employed an open-source license to enable device manufacturers to integrate and adapt the Android operating system to their devices. For example, having a permissive Apache open-source license has allowed Samsung to build its proprietary TouchWiz user interface to differentiate itself from other manufacturers. Finally, corresponding to API and SDK boundary resources aimed at software developers, Google has created the hardware abstraction layer (HAL), the compatibility definition document (CDD), and the compatibility test suite (CTS) boundary resources for device manufacturers.

Google has extended its leverage tactics even further by allowing the use of alternative app stores for the distribution of Android apps, in addition to its Google Play store. First, it has relaxed policies in its publisher terms and conditions, allowing developers to submit their apps to any app store. Second, it has incorporated more relaxed digital rights functionalities into its operating system, so that Android apps can be installed from any source and run on any Android-compatible device.

Google's policies for the use of boundary resources stand in stark contrast to Apple's approach, where developers are contractually tied to the App Store, and, under normal circumstances, iOS apps can only be sourced and installed from the App Store. In comparison to Apple, Google's leverage tactics foster an open approach that aims to grow its ecosystem as rapidly as possible. Google's core business model is focused on increasing advertising revenues, which requires as much scale as possible and hence the need to grow and leverage the ecosystem as much as possible.

This business model requires a different approach to balancing the variety vs. unity trade-off. In contrast to Apple, which tightly controls the developer side, Google has to manage the multiple sides involved with its Android platform, but gives each side more leeway than Apple does. In addition to controlling the third-party app developer

[12] For an easily understood explanation of multi-sided platforms, see: Hagiu, A. "Strategic Decisions for Multisided Platforms," *MIT Sloan Management Review* (55:2), December 19, 2013, pp. 71-80.
[13] The OHA comprises 84 technology and mobile companies that cooperate to accelerate innovation in mobile. Together, they have developed the Android platform.

ecosystem, Google must also manage its ecosystem of Android device manufacturers. To achieve this, Google uses combinations of boundary resources (e.g., in the form of HAL, CDD and CTS) to which device manufacturers must adhere. This ensures that the devices are compatible with Android and that they can run the millions of third-party apps that have been developed for it.

Google's ecosystem of third-party device manufacturers gives rise to a second control problem—fragmentation. The myriad Android devices in use do not run the same up-to-date version of Android. Contrast this with Apple's strict control of iOS integration, which ensures a much less fragmented ecosystem with a large majority of active devices running the latest version of iOS. The installed bases of different Android devices are fragmented into clusters running different versions of Android, some old and some up-to-date. This causes compatibility problems for developers who wish to use the most up-to-date Android features in the apps they build.

To address this problem, Google has developed a Google Play Services client library, which is essentially an easily updateable boundary resource that "wraps up" the APIs used by developers to build their apps. While device manufacturers are free to choose whatever version of Android they wish to install on the devices they make, Google is able to update the functionality and APIs within Google Play Services as it chooses. This enables Google to ensure that the functionality across different devices is equivalent and that developers can reach more users with up-to-date functionality in the apps they build.

With the emergence of the smart connected product era,[14] a similar fragmentation problem will confront digital platforms in other industries. Smart connected products (for example, elevators or autonomous vehicles), may suffer from version fragmentation over time. The client-library-based control tactic employed by Google provides a consistent, up-to-date and stable set of functional APIs to the platform's installed base even though they run fragmented versions of the underlying operating system. Overall, the Google case shows that a digital platform that has multiple complementor sides and massive scale should establish digital boundary resources to balance the variety vs. unity trade-off rather than use resource-intensive control tactics.

Google Rebalanced the Open Vs. Closed Trade-off for Android by Adopting Defense Tactics

Using the leverage tactic to open up a digital platform is not straightforward. A platform owner may need to make the platform open to numerous actors such as app developers, device manufacturers and app store providers, and will need to simultaneously manage and leverage each of these. Moreover, designing the boundary

[14] For a detailed description of the smart connected product concept, see Porter, M. E., and Heppelmann, J. E. "How Smart, Connected Products Are Transforming Competition," *Harvard Business Review* (92:11), November 2014, pp. 64-88

resources and policies involves making important strategic choices that have consequences impacting how other actors can use the platform and build complements on top of it. For example, choosing whether to adopt permissive or reciprocal open-source licenses,[15] or choosing to allow the use of alternative app stores can have significant consequences. While an open platform strategy benefits a business by leveraging a greater range of complementors, it can equally allow competitors to adopt exploit tactics, such as platform forking.

With platform forking, a competing platform seeks to copy, adapt and reengineer a platform to replicate it and benefit from the complementor business built on top of the original platform. For example, Amazon created the Amazon Fire platform used in its tablets by extensively exploiting Google's Android platform. First, Amazon forked Android's platform core by taking advantage of the permissive open-source license, which allowed Amazon to obtain the platform's core technology at minimal cost. This approach also enabled Amazon to benefit from Google's frequent updates to the Android platform. By using Android's open-source version control system, Amazon can take a new version of Android and merge it with the platform adaptations made since the point of the previous fork.

Second, Amazon exploited Android's complementing apps. Amazon has meticulously cloned Android's functional APIs, which developers use to build Android apps. The Google Maps API forms a significant and frequently used part of these APIs. To provide Google Maps-compatible functionality, Amazon has licensed a mapping infrastructure, a software development kit and an exact copy of the API from HERE Technologies (a mapping company). From the perspective of an app developer, this means that publishing Android apps on the Amazon Fire platform requires very little additional effort. As shown in Figure 2, this has led to roughly 500,000 developers not only distributing their apps to the Google Play store, which serves Android users, but also multihoming[16] them to the Amazon Appstore, which serves Amazon Fire users.

Google's overly relaxed publishing policies for the Android platform make such harmful multihoming possible by allowing the use of third-party app stores. In terms of deciding the appropriate degree of openness, a platform strategy thus requires a careful balance between leverage and defense tactics. To balance the open vs. closed trade-off, the platform owner can use boundary resources to defend against exploitation.[17]

To counter Amazon Fire and other platform forks, Google has developed a range of defense tactics. Earlier versions of Android bundled popular Google apps, such as Search, Calendar and Music. To prevent forked variants of Android from benefitting from these popular apps, Google has increasingly, since the "Froyo" version of Android

[15] The different types of open-source licenses are described in Table 1.

[16] For a description of multi-homing and the associated costs, see Eisenmann, T., Parker, G. and Van Alstyne, M. W., op. cit., October 2006.

[17] For detailed examples on how to use boundary resources for defensive purposes, see: Karhu, K., Gustafsson, R. and Lyytinen, K., op. cit., May 2018; and Eaton, B. D., Elaluf-Calderwood, S. Sørensen, C. and Yoo, Y., op. cit., March 2015.

in 2010, used the terms of the Apache open-source license to close further development of these apps, transforming them from open-source into closed-source programs. After Amazon entered the tablet market with Amazon Fire, Google rebranded its app store boundary resource, at that time called Android Market, into Google Play. Google effectively claimed ownership of the resource and made it clear that the app store is for the official Google Android platform and not for forked ones.

The Google Play Services client library is not merely the basis of a clever control to counter platform fragmentation across devices, but has become the basis of defense against competitors' exploitation. The speed at which Google is able to continuously develop the functionality contained within Google Play Services makes it difficult for competitors to catch up through replicating functionality in their forked variants of Android. Furthermore, by having functionality locked inside Google Play Services, Google has been able to make developers more dependent on its platform and has thus raised the barrier against multihoming their apps to competing platform forks.

Amazon and Google Focused on Exploit Tactics at the Entry Stages of their Platforms

Exploit tactics provide the means to address the critical "chicken-and-egg" problem in platform entry (where no side will join a platform without other sides already being on the platform).[18] For example, platform forking enables multihoming and thus provides access to valuable complements that will attract and grow the user base. This exploit tactic enables a platform entrant to challenge and compete against the incumbent platform, as illustrated by Amazon forking Android into its own Amazon Fire platform. The emergence of Amazon Fire resulted in Google foregoing revenue from its Android platform. Google does not benefit from app revenues from the Amazon Appstore, and it does not benefit from user data generated in Amazon Fire's ecosystem, which drives advertising revenue.

Platform forking can also be partial, with only the choicest parts of an existing platform being exploited and the rest being built separately. Google used this exploit tactic when it initially established the Android platform by selectively copying 37 fundamental packages from Java APIs. Developers were very familiar with these elements of Java and frequently used them for a variety of apps. This move helped Google lower the barrier for developers to start developing apps for Android and thus solved the chicken-and-egg problem at platform launch. Unlike Amazon, Google did not aim to exploit the complements as such, but only the skillset of the developers. Nevertheless, Google's tactic was a significant exploitation of Java, as illustrated by the

---

[18] For an understanding of the chicken-and-egg problem, see: Hagiu, A., op. cit., 2013.

ongoing $9 billion legal battle between Oracle (which owns the rights to Java) and Google.[19]

Although Apple has incorporated some parts of BSD/Unix into its iOS platform, this cannot be described as an exploit tactic, because there is no competition between the parties. Moreover, Apple keeps that particular part of the operating system open-sourced under the Darwin Project.

The exploit tactic described above involves forking and reengineering aspects of a platform. However, other exploit tactics can be used to embed a smaller platform within an existing dominant platform owned by another organization.[20] This kind of platform injection can, for example, be as harmless as the emulation of a Commodore 64 computer in an Android or iOS app. However, the threat arising from a platform injection can have significant business implications, as illustrated by the dispute between Apple and Tencent (China's most valuable company). Tencent has managed to transform its WeChat messaging app into a platform in its own right through miniprograms, which are often light versions of iOS or Android apps. Similarly, Adobe's earlier attempt to inject its Flash framework from the desktop into iOS was a tactic to exploit Apple's emerging app business by establishing a platform on top of iOS.

Challenges Involved in Balancing the Trade-offs

The unbounded nature of digital material and the ease with which a large group of uncoordinated contributors can edit and recombine it are significant factors driving the success of digital platforms. However, these factors can also lead to an explosion in both the variety of complements and in replicated efforts when many complementors independently seek to copy successful complements.

To select which complements will be made available on their platforms, platform owners need to carefully define terms and conditions for complementors and have strict approval processes. However, overly strict policies may discourage complementors or create negative publicity (e.g., criticisms of Apple's app-rejection decisions). Google has adopted a relaxed policy that allows the direct upload of any new app. But before accepting an app, it uses digital code-scanning tools to ensure that it does not contain any malware. However, initially, in the absence of a "digitized" app-approval process, Google faced the problem of weeding out malicious apps. It therefore implemented an app "kill switch," which allows the rejection of apps after they have been uploaded.

Balancing the variety vs. unity and open vs. closed trade-offs may not be entirely in a platform's own hands—national regulators, international authorities, the courts and public opinion can all influence platform policy, as well as specific decisions. In the

---

[19] For the reasoning as to why APIs cannot be copied under fair use, see the latest U.S. Appeals Court decision on the Oracle-Google lawsuit, available at http://www.cafc.uscourts.gov/sites/default/files/opinions-orders/17-1118.Opinion.3-26-2018.1.PDF.

[20] Such a platform within a platform can be characterized as a marsupial platform. For a description of this concept, see *10 Marsupial Platforms,* available at http://www.digitalinfrastructures.org/Marsupials.html.

U.S., for example, the Federal Trade Commission (FTC) intervened to protect fair competition when Apple sought to block Adobe Flash from iOS by forbidding cross-compilers. Similarly, the European Union (EU) imposed a multi-billion-dollar fine on Google for hindering fair competition by enforcing overly strict terms on how Google's services should be placed on Android devices. Also The Supreme Court of the United States decided that consumers could sue Apple in a class action suit for its monopoly over the sale of iOS apps, which can only be purchased at Apple's App Store.[21]

Balancing the open vs. closed trade-off in favor of openness may protect both the platform owner and its complementors from outside regulatory or governmental intervention. This is illustrated by the dispute between the U.S. and Huawei, which began in May 2019 when the U.S. government blacklisted Huawei.[22] As a consequence, Google (platform owner) was prohibited from licensing its services (e.g., Google Maps, Google Play and Google Play Services) to Huawei (platform complementor). For Huawei to remain in the international smartphone market, it could adopt the exploit tactic and use the openness of Android to pursue an opportunistic strategy (and in this case, one that would probably serve Google's interests as well). Importantly, it would not be sufficient for Huawei to replicate the Google Play marketplace. Instead, it could make a full platform fork (as Amazon did with Amazon Fire) and also replicate the Android APIs, including Google Maps' APIs and backend.

Four Recommendations for a Balanced Digital Platform Strategy

The four types of tactics for balancing the variety vs. unity and open vs. closed trade-offs in digital platforms strategies are executed in sequence and in combination. For example, a move to leverage a variety of complements can be balanced by subsequent control tactics to sustain the complements' compatibility and quality. Moreover, successful use of the tactics requires a combination of boundary resources. Leveraging cross-side network effects between complementors and users requires a platform owner to orchestrate APIs, SDKs, an app store, and various guidelines and terms and conditions. Similarly, an exploit tactic such as forking involves a combination of several activities, including forking the platform core and cloning the APIs. In a continuously evolving and fast-paced digital platform environment, a platform strategy evolves as it is shaped by the tactics as they are executed, enabled by a variety of boundary resources.

From our analysis of the three case platforms, we provide four recommendations for combining the digital platform tactics to build a balanced digital platform strategy. The first two are concerned with balancing the variety vs. unity trade-off, the third with balancing the open vs. closed trade-off, and the last with balancing both trade-offs.

---

[21] The Supreme Court's decision is available at https://www.supremecourt.gov/opinions/18pdf/17-204_bq7d.pdf.

[22] For the timeline of this dispute, see "The Huawei and US debacle: The story so far," *Android Authority*, March 2, 2020, available https://www.androidauthority.com/huawei-google-android-ban-988382/.

These recommendations apply not only to incumbent platform owners, but provide key guidance for other parties as well. For each recommendation, when applicable, we provide specific guidance for platform owners, complementors and new platform entrants. Furthermore, the recommendations are generalized so that they apply not only to software-based innovation platforms but also to any other digital platform involving data-, content-, or software-based complements.

1. Leverage Third-Party Complementors but Control them at Arm's Length Using Boundary Resources

The supply of a wide variety of complements is at the heart of any digital platform business model. Boundary resources, such as APIs, SDKs and monetization mechanisms, provide tools that enable and facilitate the third-party production of complements. What's more, other types of boundary resources, such as terms and conditions and guidelines, provide the rules that control what type of complements are produced.

From the platform owner's perspective, the balance between leverage and control must be orchestrated to encourage the growth of complements while preserving platform unity. Moreover, boundary resources must be designed to allow for minimal direct interaction between the platform owner and complementors. Using digital techniques, the production of complements should be leveraged and controlled at arm's length, so that production can be scaled up without exhausting the platform owner's own resources.

From a third-party complementor's perspective, control at arm's length may lead to unfair treatment by the platform owner. However, in the transparent digital world, complementors can and should protest about unfair practices. There are numerous examples of complementors contesting platform owners' decisions, which have led to better terms or better tools being made available. On the other hand, a platform owner should follow up and allow attempts to circumvent or challenge boundary resources, as they can serve as valuable learning opportunities for how best to develop the platform.

2. Manage Fragmentation When Leveraging Multiple Complementor Sides

A platform's ability to accommodate multiple complementor sides, such as device manufacturers and app developers, reflects its potential to leverage growth. However, if multiple complementor sides are interdependent, the platform may face fragmentation unless leverage is balanced with control tactics.

To avoid fragmentation, a platform owner must take control of the dependencies between the multiple complementor sides—for example, by using client-library boundary resources. These resources encapsulate the dependencies between different complementor sides and allow the dependencies to be updated frequently and independently regardless of complementors' own release schedules. As IoT and smart connected products proliferate, it will become increasingly important to manage the interdependencies between hardware and software complementors.

We recommend that complementors seeking to develop complements for digital platforms seek out platforms that are managed to avoid this type of fragmentation. Regardless of whether they are hardware or software producers, complementors can lose access to revenue-generating users when they join platforms that are prone to fragmentation.

3. Prepare for and Defend Against Exploitation by Platform Entrants

A platform that uses leverage tactics and is too open may expose itself to exploitation by competing platform entrants. If a platform both open-sources its core resources and allows alternative marketplaces for its complements, a new entrant could build a platform fork by exploiting both the core of the platform and allowing the complements to be multihomed. APIs mark key dependencies between a digital platform's complements and the core platform. By definition, APIs, must be open, but there are balancing means to protect them from being used in platform forks.

A platform owner should respond to this threat by seeking to control valuable complements and their distribution. First, a platform owner can contractually prohibit complementors from multihoming through its publisher terms and conditions, although this may prove unpopular with complementors and/or prompt regulatory actions. Second, a platform owner must carefully consider the open-source license terms for its core resources and APIs. For example, a platform owner can use dual licensing where it offers resources as open-source with stricter reciprocal terms and have separate permissive terms for partners. Third, by placing its APIs in a dynamically updateable package (see Recommendation 2), a platform owner can speed up API development and make the lives of API copycats more difficult.

A potential platform entrant can, however, seek out existing digital platforms that have not yet put measures in place to protect their cores, APIs, and complements. Such platforms make attractive targets for those seeking to create rival platforms. Exploiting digital resources from other platforms helps avoid upfront investments in establishing a new platform. Furthermore, by cloning APIs, an entrant can also encourage an incumbent's complementors to multihome, thus helping to overcome the chicken-and-egg problem of establishing an installed base of complementors and users.

Third-party complementors must be aware of the defense and exploit tactics that platform incumbents and new entrants may use. The prospect of a complementor being controlled by a platform owner to limit multihoming behavior may be unattractive, although a monopoly marketplace such as Apple's can provide superior monetization. At worst, complementors may fail to adapt to the defense tactics used by both incumbents and new entrants and see access to their apps fall away. At best, complementors can adapt and benefit from a larger installed base of users across a range of incumbent and entrant platforms.

4. Continuously Adjust Digital Platform Tactics in Response to Changes in the Competitive Landscape

Digital platform tactics are dynamic by nature and should be continuously revisited and adapted to a changing competitive environment. A tactical move made by one platform will quickly be noticed by competitors in the platform ecosystem, who will typically seek to counter the move with opposing tactics.

A platform owner should attempt to continuously balance the four tactics based on competitor moves and the stage at which the platform is in its lifecycle. The response to new exploit tactics introduced by a competitor will be a shift to defense tactics. Although leverage is important in the early stages of a platform, control and defense tactics become more important as the platform matures. Digital platform tactics also need to adapt to evolving legal constraints and interventions by regulatory bodies.

A platform entrant using exploit tactics should keep a close watch on the defensive moves by the incumbent. An incumbent may speed up API development or close up the resources completely. A platform entrant should be prepared to pace the changes, and if resources are closed up, be prepared to continue development itself or acquire the resources from another source. Understanding the licensing conditions for resources exploited is critical in planning such countermoves.

Concluding Comments

This article has identified two strategic trade-offs for a digital platform strategy and described four tactics for managing the trade-offs. Boundary resources are a platform owners' key resources for executing these tactics. Platform owners will likely be more familiar with the variety vs. unity trade-off; the need to stimulate innovation by complementors is key to platform success, but must be balanced with control tactics to preserve the unity of the platform. However, many platform owners are unaware of the tactics they can deploy to balance the open vs. closed trade-off and thus counter the threat from competing platforms. There is fierce competition for the attention of platform complementors, and there may be a significant need for a platform to rebalance the conditions in its favor.

The four tactics described in this article will help platform owners achieve the appropriate balance. The first step is to understand each of the tactics. A bigger task, however, is knowing how to use and combine the tactics to support a competitive platform strategy and achieve platform leadership.

Our recommendations provide advice on using the four tactics to create a balanced digital platform strategy. Understanding the trade-offs and having the tactics in place to balance them will equip a platform owner to make the right decisions at the right time. The increasing variety of complements can easily lead to platform fragmentation, which can be countered by using control tactics. And, openness can make a platform vulnerable to the threat of exploitation. Managing an expanding platform requires a judicious balance between control and defense tactics.

To date, the proliferation of digital platforms has been driven mostly by software-based innovation platforms or digital content marketplaces (our three cases fall into these categories). In the future, we expect industry and other data-based platforms to emerge. Data-based platforms will likely face similar trade-offs, suggesting that the digital platform tactics identified in our studies will be reusable in other contexts. Given that software is ultimately data, (i.e., lines of code), many of the principles presented here for digital innovation platforms will hold for industry data-based platforms. Digital data is as easy to modify, combine, and transfer as digital software.

Although the focus of our research and our recommendations is on digital platform owners, complementors to those platforms and aspiring platform entrants, businesses that compete within platforms rather than develop them, can benefit from understanding platform owners' strategic priorities and competitive moves. There is increasing pressure on businesses to develop a "platform play." Our research findings will also be valuable to businesses that have not yet embarked on a digital transformation and are looking for a digital business model. Exploit tactics can present an opportunity for them to bypass heavy investments and quickly augment their core operations with a digital-platform component.

Finally, the boundary between the digital and the physical has become increasingly blurred, with the rise of platform firms such as Uber and Airbnb. Although the business conducted on these platforms is physical, they rely heavily on digital technologies. The digital platform tactics set out in this article are derived from our research into pure digital platforms, but they will also be of value in emerging and rapidly evolving physical-digital hybrid contexts.

Appendix: Research Methods and Materials

This article is based on the results from our four earlier case studies on digital platform innovation listed in the table below. This work increased our understanding of the role of boundary resources in executing digital platform tactics and balancing trade-offs. In case 1, the variety vs. unity trade-off was the focus of a study both Apple iOS and Google Android. Cases 2 and 3, both on Apple iOS, also focused on the variety vs. unity trade-off. Case 4 (our most recent study) focused on the open vs. closed trade-off, and reported on Google Android and its five platform forks, including Amazon Fire. Overall, these four case studies therefore analyzed seven digital platforms.

For this article, we selected three representative digital platform examples from the seven analyzed in our original studies: Apple iOS and Google Android, the two leading (and competing) platforms; and Amazon Fire, an emerging challenger. The table below summarizes the key features of these selected case platforms. While all three platforms are digital innovation platforms, the purpose of this article is also to establish a more generalized perspective of digital platforms. We therefore refer to them mainly as digital platforms and only emphasize the innovation aspect when necessary.

Original Case Studies Used for this Article

Original Studies	Case Platforms	Trade-off
1. Tilson, D., Sørensen and C., Lyytinen, K. "Change and Control Paradoxes in Mobile Infrastructure Innovation: The Android and iOS Mobile Operating Systems Cases," *Proceedings of the Annual Hawaii International Conference on System Sciences,* pp. January 2012, 1324-1333.	Apple iOS Google Android	Variety vs. unity
2. Ghazawneh, A., and Henfridsson, O. "Balancing platform control and external contribution in third-party development: The boundary resources model," *Information Systems Journal* (23:2), March 2013, pp. 173-192.	Apple iOS	
3. Eaton, B. D., Elaluf-Calderwood, S. Sørensen, C. and Yoo, Y. "Distributed Tuning of Boundary Resources: The Case of Apple's iOS Service System," *MIS Quarterly Special Issue: Service Innovation in a Digital Age* (39:1), March 2015 pp. 217-243.	Apple iOS	
4. Karhu, K., Gustafsson, R., and Lyytinen, K. "Exploiting and Defending Open Digital Platforms with Boundary Resources: Android's Five Platform Forks," *Information Systems Research* (29:2), May 2018, pp. 479-497, available at https://doi.org/10.1287/isre.2018.0786.	Google Android CyanogenMod Xiaomi MIUI Amazon Fire Jolla Sailfish Nokia X	Open vs. closed

All four original studies employed a case study method and relied on online materials and interviews as case study evidence. In each study, we conducted extensive data collection, ranging from several hundreds to thousands of documents, covering technology news, blog posts, the case companies' websites, developer discussion boards, and legal documents including terms and conditions and open-source licenses. We used qualitative analysis to analyze the data, including coding and cross-case analysis. For details on the case study method and materials used in each study, please see the method sections in the original articles.

In preparing this article, we set out to provide a synthesized and more readable perspective for a managerial audience. With this aim in mind, we performed an interpretative synthesis[23] of our earlier case studies and the seven platforms analyzed in them, and supplemented it with recent developments around these platforms. To derive the two trade-offs and the four tactics for balancing them, we carried out a cross-case synthesis of all tactical moves in the original four cases and clustered similar moves under common categories. We then discussed and reviewed this classification and after an iterative process involving several versions, we ended up with the two trade-offs and four tactics framework illustrated in Figure 1. Because the original case articles typically

[23] For discussion on interpretative and other synthesis of case studies, see (for example) Rauch, A., van Doorn, R. and Hulsink, W. "A Qualitative Approach to Evidence–Based Entrepreneurship: Theoretical Considerations and an Example Involving Business Clusters," *Entrepreneurship Theory and Practice* (38:2), February 2014, pp. 333-368.

identified the boundary resources needed for each tactical move, it was a straightforward process to map and synthesize the key boundary resources for the four newly identified tactical groups identified for this article.

Table 4. Case Platforms Used in this Article

Case Platform	Launch Year	Key Features
Apple iOS	2007	- Following the iPhone's launch, Apple was the first smartphone firm to establish a digital platform by launching the App Store. - The marketplace and monetization mechanisms helped attract third-party developers on a large scale. - Apple exercises notably strong quality control. - Keeping the platform closed also helps prevent exploitation. - Analysis of this case shows that leveraging only one side with a focus on quality can bring sustainable success and high profits, despite restricted market share.
Google Android	2008	- Google entered the smartphone platform market after Apple and decided to use a more open approach to challenge Apple's dominance. - Instead of Apple's closed and controlled approach, Google aims to maximize leverage from two platform sides: app developers and device manufacturers. - Google cleverly employs digital boundary resources, such as Play Protect, a client library, the compatibility definition document (CDD) and the compatibility test suite (CTS), to control the multiple sides and the massive scale of complementors. - The wide openness has made Android vulnerable to exploitation by platform forks, such as Amazon Fire, and has also forced Google to actively defend its platform. - When establishing Android, Google itself exploited core APIs from Java.
Amazon Fire	2011	- Amazon Fire was a latecomer to the market, and focuses on digital content. - To leapfrog to the market, instead of building the platform from scratch, Amazon exploited the Android Open Source Project (AOSP) to get both the platform core and complementing apps from Android. - To minimize multi-homing costs for Android developers, Amazon Fire provides replicas of Android's boundary resources, including Google Maps' APIs.

About the Authors

Kimmo Karhu

Kimmo Karhu (kimmo.karhu@aalto.fi) is head of data at City of Helsinki. He is also affiliated with Aalto University, Finland, where he previously held a postdoctoral researcher position in the Department of Computer Science. Kimmo's doctoral thesis analyzed open platform strategizing and digital tactics in mobile ecosystems, and earned him a D.Sc. (Tech) from Aalto University. His current research interests include both information systems and strategic management topics such as digitalization and platform strategies. His research has been published in leading journals, including *Information Systems Research, Long Range Planning,* and *Telematics and Informatics.*

Robin Gustafsson

Robin Gustafsson (robin.gustafsson@aalto.fi) is associate professor of strategic management at the Department of Industrial Engineering and Management at Aalto University, Finland. His research focuses on strategy, organization and policy in technology-induced industry and market disruptions. His recent research explores how digital technology is disrupting existing industries and markets, platform design and strategy, sources of competitive advantage, data monetization, innovation policy in a platform economy, and agile digital strategy work. Robin is an expert in flipped classroom[24] and experienced-based learning methods and an innovative user of the Harvard case teaching method and the LEGO® Serious Play® facilitation method.[25]

Ben Eaton

Ben Eaton (be.digi@cbs.dk) is an assistant professor at the Department of Digitalisation, Copenhagen Business School, and also a visiting associate professor at the Department of Technology, Kristiania University College in Oslo, Norway. He has an MBA from INSEAD and his Ph.D., awarded by the London School of Economics and Political Science, won the 2013 ACM SIGMIS doctoral dissertation competition. Before entering academia, he had 15 years management experience in the telecoms industry. His research focuses on how organizations can establish and maintain successful digital infrastructures and digital innovation platforms and has been published in highly regarded journals, including *MIS Quarterly.*

[24] A flipped classroom is an instructional strategy and a type of blended learning that reverses the traditional learning environment by delivering instructional content, often online, outside of the classroom. It moves activities, including those that may have traditionally been considered homework, into the classroom.

[25] Developed at the Lego Group, the goal of Lego Serious Play is to improve creative thinking and communication. People build three-dimensional models of their ideas with Lego bricks and tell stories about their models.

Ola Henfridsson

Ola Henfridsson (ohenfridsson@miami.edu) is a professor of business technology at Miami Herbert Business School, University of Miami. He is also a part-time WBS Distinguished Research Environment Professor at Warwick Business School, U.K. His research and teaching focus on digital innovation, platforms and technology management. Ola's research has been published in leading journals such as *Information Systems Research*, *MIS Quarterly*, *Academy of Management Review*, and *Organization Science.* He has worked and consulted with many leading companies, including General Motors, Volvo Cars and Volvo Trucks.

Carsten Sørensen

Carsten Sørensen (c.sorensen@lse.ac.uk) is a reader (associate professor) in digital innovation in the Department of Management at The London School of Economics and Political Science. Since the 1980s, he has researched digital innovation and the innovation dynamics of mobile infrastructures and platforms. Carsten has published in leading journals, including *MIS Quarterly*, *Information Systems Research*, *Information Systems Journal*, and *Journal of Information Technology*, and has managed national, EU, and industry research projects, with research grants totaling over $3.7 million. He has also consulted with and provided executive education for many organizations, including Microsoft, Google, PA Consulting, Huawei, and Intel.