Design Rules, Volume 2: How Technology Shapes Organizations

Chapter 21 Capturing Value via Strategic Bottlenecks in Digital Exchange Platforms

By Carliss Y. Baldwin

Note to Readers: This is a draft of Chapter 21 of *Design Rules, Volume 2: How Technology Shapes Organizations*. It builds on prior chapters, but I believe it is possible to read this chapter on a stand-alone basis. The chapter may be cited as:

Baldwin, C. Y. (2021) "Chapter 21—Capturing Value via Strategic Bottlenecks in Digital Exchange Platforms," *Design Rules, Volume 2: How Technology Shapes Organizations*. Harvard Business School Working Paper (August 2021).

I would be most grateful for your comments on any aspect of this chapter! Thank you in advance, Carliss.

Abstract

The purpose of this chapter is to use value structure analysis to better understand how sponsors of digital exchange platforms capture value and maintain strategic bottlenecks. I begin by describing the three core technological processes that lie at the heart of all digital exchange platforms: (1) search and ad placement; (2) dynamic pricing; and (3) data analysis and prediction. These processes are carried out automatically and at high speed by computerized algorithms. The algorithms consist of steps, all of which are essential to the successful completion of a transaction or transmission of a message. However, bottlenecks are inevitable any synchronized multi-step flow process. The platform sponsor's first responsibility is thus to resolve bottlenecks in its core processes when and where they arise.

One of an exchange platform's critical core processes is to generate predictions as to what the user would like to see and/or purchase. Better predictions speed up the flow and reduce the unproductive time users spend on the platform. Thus to be competitive today, platform sponsors must have the ability to gather data, generate predictions and test hypotheses within time intervals measured in milliseconds. Predictions can be improved by tracking individuals' behavior online, but only at the cost of reducing their privacy. The inevitable tension between performance and privacy has yet to be resolved.

The chapter goes on to discuss how platform sponsors can "lock-in" members of an ecosystem through a combination of visible instructions, data storage, and specific inertia. It concludes by describing successful and unsuccessful attempts to disintermediate exchange platforms.

Introduction

In Chapter 7 I argued that the key to capturing value in any large technical system is to control an essential and unique functional component— a strategic bottleneck. In open platform systems, a strategic bottleneck may be achieved by (1) controlling visible instructions needed to achieve interoperability between complements of the platform; (2) addressing flow bottlenecks in the core platform processes; and (3) avoiding disintermediation.

In standards-based platforms, bottlenecks arise from ownership of visible instructions (standards). In contrast, exchange platforms must pay more attention to the the speed at which transactions and messages cross their sites. However, we shall see that control of visible instructions is a means of achieving "lock-in" and protecting the platform sponsor from disintermediation.

The purpose of this chapter is to use value structure analysis to better understand how sponsors of digital exchange platforms capture value and maintain strategic bottlenecks. I begin by describing the three core technological processes that lie at the heart of all digital exchange platforms: (1) search and ad placement; (2) dynamic pricing; and (3) data analysis and prediction. These processes are carried out automatically and at high speed by computerized algorithms. The algorithms consist of steps, all of which are essential to the successful completion of a transaction or transmission of a message. However, bottlenecks are inevitable in any synchronized multi-step flow process. The platform sponsor's first responsibility is thus to resolve bottlenecks in its core processes when and where they arise.

One of an exchange platform's critical core processes is to generate predictions as to what the user would like to see and/or purchase. Better predictions speed up the flow and reduce the unproductive time users spend on the platform. Thus to be competitive today, platform sponsors must have the ability to gather data, generate predictions and test hypotheses within time intervals measured in milliseconds. Predictions can be improved by tracking individuals' behavior online, but only at the cost of reducing their privacy. The inevitable tension between performance and privacy has yet to be resolved.

The chapter goes on to discuss how platform sponsors can "lock-in" members of an ecosystem through a combination of visible instructions, data storage, and specific inertia. It concludes by describing successful and unsuccessful attempts to disintermediate exchange platforms.

21.1 Flow Processes at the Core of Digital Exchange Platforms

All digital exchange platforms contain a common core of flow processes performed automatically by machines without human intervention. The absence of humans in the core flow processes permits the processes to attain levels of speed and capacity impossible in any previous production system. Performance also improves over time due to the dynamics of Moore's Law and advances in computer architecture and programming (for example, parallel processing).

On a digital exchange platform, the core flow process begins when a user initiates an interaction by entering a request or landing on the platform's website. On a *transaction* platform, a buyer's query is implicitly a request to search and view ads for products he or she might want to buy. The platform responds by dynamically loading a set of icons with abbreviated information about products and links to pages with more information.

Prices are a key part of the product description. The prices may be set once and for all, but digital transaction platforms are capable of changing prices rapidly based on transient shifts in supply and demand. (Recall Dell's practice of "demand shaping" described in Chapter 19.) Very frequent changes must be managed algorithmically using programs provided by the platform.

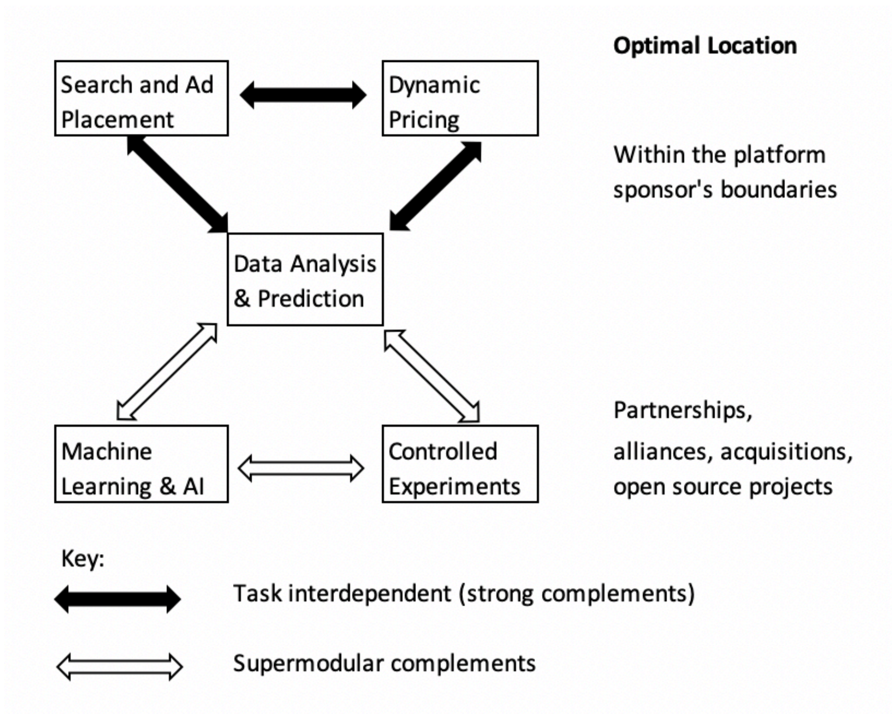
On a *communication* platform, the initial request is not to view products but to search for senders or receivers of messages. In addition, the majority of communication platforms today are supported by advertising revenue. Thus while the platform is fulfilling the users' primary request for messages, it must simultaneously select, place and price advertisements. The ads must appear in the time it takes the platform to reload a page, i.e. in a matter of milliseconds. And just as products are priced dynamically, ads are priced dynamically via automated auctions among advertisers.

Thus search and ad placement and dynamic pricing are core processes common to all transaction and communications platforms. On a large digital exchange platform, millions, perhaps billions of these automated, dyamically managed events take place on the site every day. Synchronized algorithms are executed in parallel at a massive scale as part of the platform's everyday functioning.

Both ad placement and product pricing depend on the platform's ability to predict user preferences and influence user behavior. The platform must marshal relevant data and form a prediction as to the best way to respond to the user's query. The process of predicting is the third core process that runs in real time on digital exchange platforms. (The platform may have other processes running in parallel as well. For example, communication platforms must simultaneously deal with the user's primary request.)

The three core processes—search and ad placement, dynamic pricing, and data analysis and prediction—are depicted in the top half of Figure 21-1. As indicated by the black arrows, within any query, the three processes are interdependent and therefore strong complements. To function in a timely fashion, the algorithms must pass information back and forth. A delay at any point will delay subsequent steps and may effect the user's experience.

Figure 21-1 Technologies Needed by Digital Exchange Platforms



Flow bottlenecks are endemic to interdependent step processes.[1] In a digital exchange platform, bottlenecks can arise because of the coding of instructions, the configuration of hardware, the bandwidth of communication channels. As a Google engineer observed:

> Your site is slow, what do you do? What resource is being restrained? Because it's always a *bottleneck*. Is it CPU-bound? Is it disk-bound? You have to understand the fundamentals of what makes things fast or slow.[2]

The bottlenecks are addressed by the modern equivalent of systematic management—the close analysis and continuous redesign of interdependent steps. Managers and engineers must be able to range over the entire system and take timely action to rearrange steps, modify code, and set new protocols and boundaries. Such synchronization requires high levels of transparency and consistent priorities—conditions that, I have argued, are most easily achieved within a single firm via unified governance, direct authority and hierarchical decision-making.[3]

---

[1] The propositions derived in Chapter 8 for interdependent step processes apply to the platform's core algorithms. First, adding steps to a process increases the probability of creating a new bottleneck in the system. Second, increasing the variability in execution time for any instruction reduces expected throughput for the system as a whole. The second principle is a core tenet for RISC architectures in the design of microprocessors. Hennessy and Patterson (1990).

[2] Pete Buchheit as quoted by Levy (2011) p. 167. Emphasis in original.

[3] Open source projects have demonstrated that organizations subject to distributed governance can create high-

The interdependence of the core algorithms in digital exchange platforms makes the underlying technologies strong complements: none will work except in the presence of the other two. Strong complementarity in turn makes it difficult to split the activities among multiple organizations linked by transactions and contracts.[4] There are no thin crossing points in the algorithmic task network, thus the tasks are best organized within a single company.

As we saw in Chapter 20, in 2020, the sponsors of most digital exchange platforms were *profit-seeking corporations* under unified governance. Organizations subject to distributed governance, such as open source communities, generally do not have the ability to act quickly and decisively in response to real-time events. Non-profit corporations do not have strong incentives to pursue efficiency or growth as paramount goals. In contrast, for-profit corporations benefit directly from increases in efficiency due to the removal of bottlenecks, and with higher levels of profitability (higher *ROIC*), they can grow faster. Under unified governance, they can use direct authority and set up managerial hierarchies to pursue the goals of profitability and growth.[5]

21.2 Improving Predictions

The bottom half of Figure 21-1 shows the technologies needed to support state-of-the-art data analysis and prediction. Machine learning is a set of statistical tools and algorithms designed to automatically formulate and test fine-grained hypotheses on very large data sets. Over time, the algorithm "learns" to improve its model. For example, an algorithm might predict the probability of a sale as a function of hundreds of variables related to the product, the user's demographics and peer group, and the user's own past patterns of viewing and purchase. Products with the highest predictive scores are then shown to the user within milliseconds of the user's arrival on the platform. The algorithm then records the user's response and recalibrates the prediction model.

Artificial intelligence (AI) is a related set of computational tools, through which computers learn to interpret non-numerical data, such as natural language text, speech, images, and convert the primary data into valid responses and predictions. Machines can then respond to informal, non-specific and ambiguous inputs in ways that humans find understandable and appropriate.

Finally, digital platforms use real-time experiments to test the validity and power of their predictions. For example, if predictive models suggest that two page designs are equally likely to please viewers, the platform might test both. This practice, known as A/B experimentation is similar to running a set of controlled laboratory experiments.[6] However, because the experiments

---

quality code, but such organizations do not attempt to tightly time their members' activities. They are the focus of Chapters 23-25 below.

[4] Hart and Moore (1990).

[5] Iansiti and Lakhani (2020) pp. 79-183 describe in some detail the creation of a set of interdependent but flexible core processes at Amazon and Microsoft.

[6] Tang et al. (2010); Kohhavi and Thomke (2017).

are electronic, they are both fast and cheap. They are used to increase the power and accuracy of platform predictions.

Data analysis and prediction, machine learning and AI, and controlled experiments each have value in their own right, thus are not strong complements. However, these technologies are supermodular complements in the sense that improvements in one make the others more valuable. More data provides larger training sets for machine learning and artificial intelligence and more opportunities for controlled experiments. Predictions from statistical models can in turn be combined with experimental results, leading to better predictions on the next round. The new predictive models can then be used to respond to actual queries coming in from the real world, generating still more data. Better predictions lead to better ad placements and pricing decisions, fulfilling the upper group of complementarities. The end result is to improve users' experience on the platform and increase the platform's revenue.

The technologies in the lower half of Figure 21-1 can be separated from the platform's core processes. The platform sponsor thus can decide which lines of inquiry to pursue inhouse and which to source externally, using the tools of open innovation.[7] For external sourcing, it may enter into partnerships and alliances with universities and other companies, and join open source projects and research consortia. Finally, it can monitor early-stage companies, with the goal of acquiring complementary technologies that have the potential to become strategic bottlenecks in their own right. Distributed modular complementarity (DMC) holds for these activities as evidenced by the large ecosystem that has grown up around machine learning, AI and decision making based on prediction and experiments.[8]

21.3 Personal Data Tracking and Privacy

In the digital world, every action leaves a trace. The power and capacity of today's computers mean that ephemeral traces can now be stored, manipulated, and analyzed. They become data.

An exchange platform sits in the center of a flow of transactions and/or messages. If its managers choose (and virtually all do), the platform can monitor the flows, store its metrics, package the data, and use it to improve the core processes of ad placement and pricing. As Google discovered in 2001, scraping users' data from the electronic feeds and subjecting it to sophisticated predictive analysis, can greatly improve the efficacy of ad placement, and thus increase revenue.[9]

The granularity of the platform's data extends down to the level of the individual user. The platform "knows" who is accessing its site—sometimes by name as with Facebook and Gmail, sometimes by IP address as with Google Search. The algorithms can use accumulated data about a user's searches, purchases, clickthroughs, posts, emails, etc. to improve its

[7] Chesbrough (2003; 2006); West and Bogers (2014).

[8] Jacobides, Brusoni, Candelon (2021).

[9] Levy (2011); Wu (2017).

predictions and change what the user sees. The purpose of these manipulations is ultimately to influence the user's actions in ways that have a positive impact on the platform's revenues.

Personal data tracking interferes with privacy and thus is a controversial practice. On the one hand, by enabling the platform to perform better matches in real time, the practice improves the user's experience and reduces the platform's costs. On the other hand, manipulation of unconscious attitudes and impulsive behavior smacks of mind control: It diminishes the user's autonomy and compromises his or her free will.[10] The EU has enacted restrictions on these practices, and other jurisdictions including China and California are considering similar measures.

How effective is online advertising informed by personal data tracking? First, we have data about the spread of Internet advertising and the degree to which it has replaced other advertising channels. Figure 21-2 shows total ad spending in the U.S. for all media from 2001-2018, broken down into non-Internet, desktop Internet, and mobile Internet categories. Essentially all the growth in advertising spending from 2006 to 2018 came from Internet advertising. In contrast, non-Internet ad spending dropped substantially from a peak of $156 billion in 2006 to $116 billion in 2018. (As a percent of GDP, total advertising declined from 1.3% in 2001 to 1.09% in 2018. Advertising has been a low-growth industry for most of the past decade-and-a-half.)

Some of the displacement of traditional advertising channels by Internet channels may be attributable to the effectiveness of personal data tracking. However, many other cultural and demographic factors were at work in the same time interval, hence this evidence is suggestive but not conclusive.

Published studies of the effectiveness of personal data tracking are scarce, but their results are quite consistent. *On both communication and transaction platforms, using personal tracking data to improve ad placements has a large positive impact on the effectiveness of the ads*. For example, the latest round of EU privacy regulations (restricting personal tracking) was associated with a 10% decline in page views and a 5.6% decline in orders placed on e-commerce sites.[11] Confirming these results, in ad auctions, bids for ads shown to consumers who do not allow personal tracking are around 50% lower than bids for ads to consumers who do allow personal data tracking.[12]

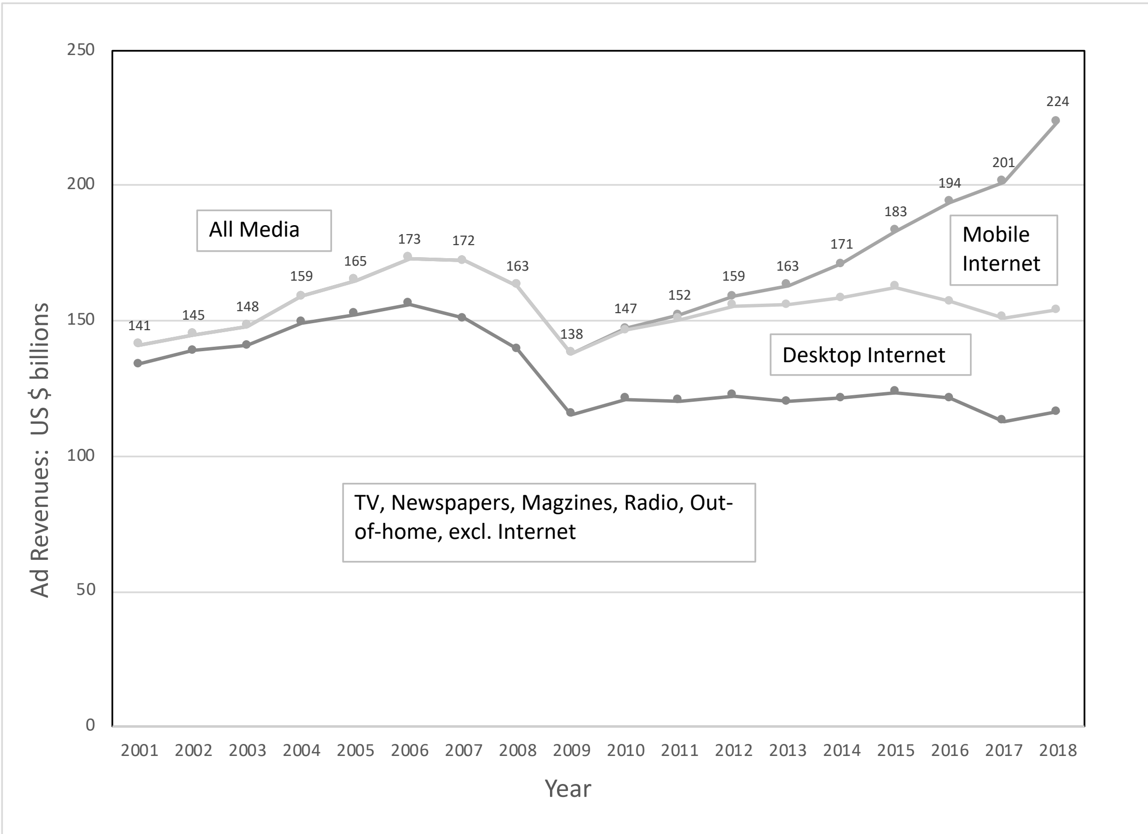
---

[10] Zuboff (2019).

[11] Goldberg, Johnson and Shriver (2019). An earlier study by Goldfarb and Tucker (2011) of the first privacy directive in the EU found similar results.

[12] Johnson, Shriver and Du (2020).

Figure 21-2 U.S. Spending on Advertising by Channel 2001-2018



Sources: The author based on data from: IAB/PwC Internet Ad Revenue Report, Full Year 2007, 2012, 2018; LEK Consulting, Future of Advertising Spend, September 2015: https://www.lek.com/sites/default/files/insights/pdf-attachments/Future-Of-Advertising-Spend_ExecutiveInsights_Spotlight4.pdf (viewed 1/15/20);

Statista, Media Advertising Spending in the United States from 2015-2022, December 2018: https://www.statista.com/statistics/272314/advertising-spending-in-the-us/ (viewed 1/15/20).

Although many people espouse concerns about online privacy, in the U.S. at least, the number of people who make an effort to "opt-out" of personal data tracking is miniscule: on the order of .16%. Other forms of ad-blocking, for example, by browsers, may be more effective. In any case, most web users appear to accept the practice of personal data tracking and will curtail their viewing when platforms do not provide them with personalized matches.[13]

[13] Sun et al (2019); Goldberg, Johnson and Shriver (2019).

21.4 Locking in Users through Visible Instructions, Data Storage, and Specific Inertia

Superior control of core platform processes can make a digital exchange platform into a strategic bottleneck through greater speed and better tracking, which reduces users' wasted time. However, speed and predictive ability do not by themselves affect a user's cost of switching to another platform. This section considers the ways in which a platform sponsor can reinforce its strategic bottleneck through control over visible instructions, storage of data, and specific inertia.

Visible Instructions (APIs)

Control of visible instructions is the defining property of a standards-based platform (see Chapter 14). Efficiency of core processes is a secondary concern.[14] In contrast, the defining property of a digital exchange platform is the ability to complete transactions or transfer messages. However, if at the same time visible instructions owned by the exchange platform become embedded in users' coded programs, the instructions will increase those users' switching costs. As a result, it is not surprising that many digital exchange platforms are also standards-based platforms. They publish APIs and encourage their clients (on different sides) to use them. Clients who must rewrite software or redesign systems in order to switch to another platform will then have reason to stay put.

Table 21-1 reproduces the list of "top twenty" *subsidiary* standards-based platforms in the Apple iPhone's ecosystem between February 2008 and August 2016 (see Table 16-2). The subsidiary platforms control APIs, which then are embedded in the code of iPhone applications. For example, if an app developer wishes to show a map, they can use GoogleMaps APIs. To edit Instagram photos within the app, a user can access applications using Instagram APIs.[15]

I have added four columns to the original table. They show (1) whether the sponsor of the subsidiary platform was a digital exchange platform; and (2) if so, the ultimate owner of the digital exchange platform; (3) whether the exchange platform provided data storage in addition to transactions and communication services and APIs; and (4) whether the exchange platform had high or low "specific inertia" (defined below).

Not surprisingly, the vast majority of the "top twenty" publishers of APIs in the iPhone ecosystem during the period of observation were digital exchange platforms. Facebook, Twitter and others are not only transaction and communication hubs, but also standards-based platforms that encourage complementors to create applications that work on their platforms using their APIs.

---

[14] Evidence of the pre-eminence of visible instructions can be found in Intel's decision in the mid-1990s not to embrace RISC-based architectures, which were faster, but would break the chips' compatibility with prior instructions. See Inset Box 12-1. Grove (1996) pp. 103-106.

[15] In April 2018, at the height of the Cambridge Analytica scandal, Instagram abruptly and without announcement, shut down its "public" API platform, three months before the announced end date. This had the effect of "breaking" many applications tailored to that platform. The old API functions have been partially replaced by a new Instagram Graph API. Smith (2018).

Table 21-1 "Top Twenty" Subsidiary Platforms in the Iphone Ecosystem

	Platform Application	No. Dependent Applications	Digital Exchange Platform	Owned by	Stores User Data	Specific Inertia
1	Facebook	76845	√	Facebook	yes	high
2	Twitter	60703	√	Twitter	yes	high
3	YouTube	20009	√	Google	yes	high
4	Gmail	9851	√	Google	yes†	low
5	Instagram	8223	√	Facebook	yes	high
6	Dropbox	4872	√	Dropbox	yes	low
7	Vine*	3025	√	Twitter	n.a.	n.a.
8	GoogleMaps	2627	√	Google	yes†	low
9	Uber	2331	√	Uber	no	low
10	Yahoo	2093	√	Verizon Media	yes	high
11	Flashlight	2081				
12	Amazon	1980	√	Amazon	yes††	low
13	Tumblr**	1895	√	Automattic	yes	high
14	Photoeditor	1695				
15	WhatsApp	1644	√	Facebook	yes	low
16	Pinterest	1327	√	Pinterest	yes	high
17	LinkedIn	1275	√	Microsoft	yes	high
18	Chrome	1215				
19	Indeed	1202	√	Recruit Co., Ltd.	yes	low
20	Google	1129	√	Google	yes†	low

* Website shut down, January 2017.
https://www.wired.co.uk/article/vine-return-camera-app (viewed 12/29/19).
** In August 2019, Verizon Media sold Tumblr to Automattic, an ad-supported web publishing and website hosting company.
† Google storage offered through Google Drive.
†† Amazon storage offered through Amazon Drive.

Source: Shiva Agarwal, private communication. The platform applications are ranked according to the number of applications entering the Apps Store between August 2008 and February 2016 that were linked to the subsidiary platforms via APIs. For a description of the dataset, see Kapoor and Agarwal (2017).

Data Storage

It is easier to move digital files than to remove unique APIs threaded through coded programs. Even so, it is not a move to be taken lightly. As a result, stored data, in addition to visible instructions, is a second source of user lock-in for digital exchange platforms.

Before the advent of cloud computing, users stored their digital possessions on devices (mostly disks) they purchased and managed themselves. Today those digital files have mostly migrated to remote servers owned and managed by digital exchange platforms.

The migration was in many ways a natural consequence of the platforms' own investment trajectories combined with economies of scale. The platforms' core processes are hungry consumers of storage capacity. Storage hardware and software were subject to classical economies of scale (bigger is cheaper) *and* Moore's Law (newer is cheaper). As the exchange platforms added storage capacity to address bottlenecks in their core processes, their unit cost of storage dropped, and they became low-cost producers. For example, Amazon Web Services grew out of Amazon's work to build information technology infrastructure at scale to support Amazon's high growth rate in transactions and logistics.[16]

The platforms found that they could expand their scale and further decrease their cost, by providing remote storage to third parties. They could also use free storage to enhance other products such as email. (Google's Gmail originally offered much more storage than its main competitors Microsoft's Hotmail and Yahoo!Mail.)

With lower costs, and the increasing ubiquity of communication networks, local control over digital possessions soon became an expensive and unnecessary luxury.

Table 21-1 shows that most of the large digital exchange platforms also offered remote "cloud" storage and other services to their users. The one exception, Uber, provides only transient matches between riders and drivers, thus long-term data storage bears almost no relation to its main value proposition.

Responding to these trends, in October 2019, three U.S. senators introduced legislation (the ACCESS Act) aimed at increasing users' ability to switch among platforms by making user data portable and platforms more "interoperable." Earlier that year, a number of large technology companies, including Facebook, Google and Apple, launched the open source Data Transfer Project that would allow individuals to move data between different platforms easily. However, as of 2021 both the legislation and the project appear dormant.[17]

Specific Inertia

The third source of lock-in arises from network effects. Members of a platform's ecosystem—the sides—are, by definition, *complements of the platform and of each other*. This is the essence of network effects, which are a defining characteristic of all open platform systems (see Chapter 11). At the same time, members of the ecosystem are, by definition, autonomous agents. Each member thus makes key decisions, including decisions on whether to use the platform, independently of the others.

The question is, do users (on any side) care about the *specific identities* of members on their own or another side? Or do they only care about the platform's generic attributes, like the size of the installed base? If users care only about generic attributes, then a rival platform can mount a challenge by matching those attributes. However, if users care about specific identities (friends and family, for example), then members of the ecosystem face a severe collective action

[16] Miller (2016).

[17] Robertson, (2019). As of mid-2021, no action has been taken on the bill since it was referred to committee on October 22, 2019. https://www.congress.gov/bill/116th-congress/senate-bill/2658/all-info (viewed 8/16/21).

problem. Moving to a different platform makes sense for one member only if *identified others* move as well. The rival platform may have no way to know who "the others" are and no means to organize the sides into coherent blocks that can take group action. In this case, members of the ecosystem are locked-in by what I am calling "specific inertia."

An example may serve to clarify the difference between specific and generic inertia. On ride-sharing services, such as Uber and Lyft, riders don't care about the presence of particular drivers ("Maria's cousin"). They only care about the drivers' generic attributes: their ability to drive, the condition of the vehicles, politeness, and safety. These generic requirements can be addressed by background checks and user reviews. After that, the riders care only about waiting time and prices, which in turn depend on the total number of drivers. Drivers, conversely, do not join the platform to transport specific people ("Mike's grandmother"), but generic passengers. If a rival platform can come close to matching the generic attributes of the incumbent's platform, drivers and riders will have no compunction about switching from one platform to the other, regardless of what their peers do. These platforms have low specific inertia.

Other platforms with low specific inertia include search engines, modern email services, mapping services, and many retail sites. (In the language of platform strategy, these are "multihoming" platform systems.) For example, with search engines, searchers, sites and advertisers mostly care about overall coverage, speed, and the predictive accuracy of the algorithms, rather than the presence/absence of specific users or sites.

Contrast these examples with social media platforms such as Facebook, Twitter, and Instagram. Here the primary users expect to connect with specific people via the platform. If half one's social network is on one platform and half on another, and the platforms don't interoperate, the entire system is worth much less than if all are on one or the other platform. Even if some would prefer the generic attributes of another platform, the presence of specific other parties can offset the current platform's generic disadvantages. User lock-in is higher for platforms with specific inertia than those with generic inertia.

In summary, three factors operate to increase users' switching cost, thus reinforcing an exchange platform's status as a strategic bottleneck. First, like standards-based platforms, digital exchange platforms may publish visible instructions (APIs) that become woven into programs created by developers and users. In fact, the largest publishers of APIs in the Apple iPhone ecosystem are digital exchange platforms. Second, digital exchange platforms have become low-cost providers of storage capacity to users ranging from individuals to large corporations. Most large digital exchange platforms provide some amount of free data storage services to their users.

Finally, users on some platforms care about the presence of other specific users (corporations or individuals) on the same platform. Unlike visible instructions and stored data, which create bonds between the user and the platform, specific inertia creates *ties among specific users*. Because groups, not individual agents, are locked-in to the platform, platforms with high levels of specific inertia may continue to be active long after their core processes have become inefficient and obsolete.

21.5 Platform Disintermediation

Investing in superior core processes and increasing user switching costs are ways in which a platform sponsor can achieve and defend the platform's status as a strategic bottleneck. However, as we saw in Chapter 16, every strategic bottleneck is subject to the threat of distintermediation. Recall that disintermediation can arise in four ways:

- by substitution;
- via reverse engineering;
- through the creation of platform-independent complements;
- through envelopment (offering bundled platforms at a cost below that of a competing stand-alone platform).

In the next sections, I describe examples of all four types of disintermediation of digital exchange platforms. In these encounters, the exchange platforms have been both targets and attackers.

21.6 Distintermediation by Substitution

The most salient example of disintermediation by substitution in the late 20th century was the migration of commerce and communication to the Internet. Beginning in the mid-1990s, digital exchange platforms provided a convenient alternative to physical distribution channels and a cheaper way to send and receive information. Among the casualties of the Internet were three mass market online platforms, Compuserve, Prodigy and America Online (AOL). Their's was a classic disintermediation, involving the substitution of a high-cost, inefficient conduit with a faster, cheaper substitute.[18]

The business model for the original online services was a monthly subscription fee plus charges for additional connection time.[19] The basic connection method was a modem and a dial-up line to the company's servers, which formed a closed proprietary network. Slow response time, busy signals, and crashes were a common occurrence made more infuriating by the fact that users were charged on the basis of time connected. Finally, because most media and entertainment channels and shopping outlets had no online presence, most of the content was prepared by the platforms themselves.

In the mid-1990s, the World Wide Web (WWW) emerged as a non-proprietary set of

---

[18] The term "disintermediation" was coined in the financial services sector in the 1980s to describe the movement of investors from "intermediated" investments such as bank, trust or brokerage accounts to more direct investments in mutual funds, which charged much lower fees. Disintermediation of transaction platforms via direct dealing between buyers and sellers has been the focus of studies by Gu (2021) and Gu and Zhu (2021).

[19] Lewis (1993). Prodigy attempted to rely on ad revenue and transaction fees for most of its revenue, but was not able to turn a profit on these sources alone. It then attempted to charge fees on email messages and connections to message boards, but backed down when faced by wholesale defections. Prodigy (online service) *Wikipedia* https://en.wikipedia.org/wiki/Prodigy_(online_service) (viewed 2/19/20).

standards for linking computers via the Internet. The first "website" appeared in 1991.[20] In 1993, the National Center for Supercomputing Applications introduced Mosaic, a web resource locator ("browser") with a point-and-click graphical user interface.[21] From that point, the web exploded in all directions: by the middle of 1996, there were approximately 230,000 websites, almost 70% of them commercial.[22] In an outburst of peer production, Internet users created vast amounts of web content that was freely available to other users.[23]

Users of Compuserve, Prodigy and AOL were then faced with the choice whether to access the web via an expensive, often slow subscription service or a free, easy-to-use browser and a cheap Internet Service Provider.[24] Individuals and companies left the older online services by the millions.

A second example of disintermediation by substitution is the case of Yahoo. Founded in 1994, as the web was taking off, Yahoo's distinctive offering was the Yahoo Directory, a searchable listing of web pages organized hierarchically by topic. The original directory was compiled by human beings, then augmented by early web crawlers and search engines. In 1998, Yahoo was the most popular starting point for users of the Internet, with approximately 95 million page views a day and 30 million unique users.[25] However, the hierarchical classification scheme of the Yahoo Directory proved impossible to scale as the number of websites expanded in the late 1990s.

Google's search engine, launched in 1999, proved a superior substitute for the Yahoo Directory. The Page Rank algorithm, which counts the number of links to a page, was combined with other carefully weighted "signals" to create *a fully-automated and scalable core process at the heart of Google's search engine*.[26]

Famously, Yahoo's founders declined an offer to license the Page Rank algorithm in 1998. They failed to realize that automated search and ad placement were critical flow processes for their platform. In 1998, search was a technical bottleneck affecting every user of the Internet. By solving the technical bottleneck with a proprietary algorithm, Google transformed it into a strategic bottleneck—essential, unique and owned by Google.

Today Compuserve, AOL, and Yahoo are all owned by the telephone company, Verizon.

---

[20] Gray (1996).

[21] Marc Andreesen, who led the team that created Mosaic, went on to found Netscape, the target of Microsoft's browser envelopment move in 1996. See Chapter 17.

[22] In January 2020, there 170,000 were active websites. This number has been fairly stable since 2011. Netcraft (2020).

[23] Greenstein (2015).

[24] Internet Service Providers (ISPs) were commercial or community-backed ventures that provided basic Internet access for a monthly fee. There were few barriers to entry, thus subscription fees were low. Today in the U.S., Internet access is part of basic telephone or cable television service.

[25] BBC News (1998).

[26] Levy (2011) pp. 40-56.

(Prodigy became an Internet Service Provider, but eventually was shut down.[27]) Each platform maintained a separate website, *but the core processes of ad placement, ad pricing, data analysis and email are centrally managed by the parent company*.[28] Integrating the platforms' core processes provided economies of scale while maintaining the distinctiveness of the individual brands.[29] In January 2020, Verizon added a search engine to its centralized services: a transaction platform was in the works.[30]

What lessons can be drawn from these cases of disintermediation by substitution?

First, *inferior core processes were the root cause of disintermediation*. The Internet and World Wide Web were almost free substitutes for Compuserve, Prodigy and AOL's proprietary networks and content. In the case of Yahoo, Google's search capabilities were far superior to Yahoo's directory. Google's email service, Gmail, offered an alternative to Yahoo mail with less intrusive ads and more storage. Last but not least, Google's automated ad placement auctions were a truly revolutionary substitute for the former practice of selling ads person-to-person via a salesforce.[31]

Second, *supermodular network effects work in both directions.* When an exchange platform's ecosystem is growing, more users on one side attract users on other sides. However, if the platform's ecosystem begins to shrink, users with low switching costs will be on the lookout for substitutes. Substitutions by one side in turn will stimulate substitutions by other sides. The platform's ecosystem will contract, reducing economies of scale for the core processes. A vicious cycle follows until the inferior platform disappears or is enveloped by a larger platform organization that can restore economies of scale.

Third, *user lock-in slows down and may even prevent a platform's collapse when confronted by a superior substitute*.[32] Long after their heyday, Compuserve, AOL and Yahoo all survived. Each was kept alive by a persistent, albeit shrinking, ecosystem of loyal users.

21.7 Disintermediation by Reverse Engineering—The Dog that Did Not Bark

The most significant examples of reverse engineering affecting digital exchange platforms were those that did not happen. Recall Scott Mueller's explanation of why Apple's Macintosh escaped reverse engineering:

> [The Macintosh] BIOS is very large and complex and is essentially part of the OS, unlike the much simpler and more easily duplicated BIOS found on PCs. The greater complexity

[27] Prodigy (online service) *Wikipedia* https://en.wikipedia.org/wiki/Prodigy_(online_service) (viewed 2/19/20).

[28] Verizon Media Homepage, https://www.verizonmedia.com/ (viewed 1/2/20).

[29] In fact, the strategy is identical to Alfred Sloan's strategy of offering different makes of car built from a common set of underlying components—see Chapter 10.

[30] Toh (2019).

[31] Levy (2011) pp. 83-120.

[32] Arthur (1989); Zhu and Iansiti (2012).

and integration has allowed both the Mac BIOS and OS to escape any clean-room duplication efforts.[33]

In other words, the best defense against reverse engineering is complexity derived from the integration of functions into interdependent processes. We saw this principle at work twice in Chapter 17. First, partly to delay reverse engineering by AMD, Intel used proprietary internal buses to integrate the chipset and processor. Second, Microsoft made Windows ever larger and more complex by enveloping previously stand-alone functions into the operating system itself. Today, Intel chips and Windows have too many internal dependencies and undocumented functions to allow perfect reverse engineering. (Open source efforts to implement Windows lag far behind the the latest versions of the system.)

When developing the iPhone, Apple went one step further by integrating the entire "vertical stack" —chip, hardware, and operating system.[34] This high degree of integration served the dual purpose of making core processes more efficient and reverse engineering more difficult.

Early digital exchange platforms thus had to depend on standards-based platforms that *were effectively immune to reverse engineering*. To access a digital exchange platform, users first had to purchase a computer or smartphone with an operating system. The profits captured by the hosts raised the users' cost of owning the system, yet did not flow to the exchange platforms. In the next chapter we will see how Google bypassed these strategic bottlenecks in hardware and software by creating an open source operating system, Android.

The integration of platform functions to improve performance and deter reverse engineering continues to be a common technological move. In January 2019, *The New York Times* reported that Facebook planned to integrate its three messaging services—WhatsApp, Instagram messaging, and Facebook Messenger—into a single technologically unified codebase. (WhatsApp and Instagram were originally separate companies, thus their codebases are currently independent of each other and Facebook Messenger.)[35]

On the one hand, creating a single messaging service would allow users of different services to communicate more easily, and reduce Facebook's costs of maintaining and developing new features across multiple codebases. It would also make it easier to achieve Facebook's avowed goal of bringing end-to-end encryption to all users. On the other hand, the new codebase was likely to have more features and be more complex than the three legacy codebases, thus it would be harder for a third party to compete by reverse engineering those functions. It might also make it more difficult to break Facebook apart into separate companies.[36]

---

[33] Mueller (2005) p. 28, quoted in Chapter 15.

[34] Malik (2019). In the Wintel ecosystem, the microprocessor and chipset, the computer, and the operating system were three distinct modules.

[35] Isaac (2019).

[36] This argument is suspect because, if a breakup were ordered, all three descendents could be given copies of their common infrastructure code, and/or create an open source project to maintain it. See the discussion of Android

For this reason, in 2019, the FTC considered filing a preliminary injunction to block technological integration of these services.[37] In December 2020, the Federal Trade Commission and 48 state attorneys general brought lawsuits against Facebook, charging antitrust violations and asking the court to unwind the acquisitions of Instagram and WhatsApp, spinning them off into separate companies.[38] The lawsuits were dismissed in June 2021, but the FTC refiled its suit with additional arguments and evidence in August.[39]

21.8 Platform-Independent Complements—Web Applications

With the main standards-based platforms effectively immune to reverse engineering, digital exchange platforms had to fall back on the strategy of making their applications platform-independent. This was first accomplished using web pages, then "web apps". Instead of using the visible instructions (APIs) of the operating system, web applications use visible instructions (protocols and languages) defined for the web. The location and formatting of these pages depended on the World Wide Web protocols and the HTML computer language, not on Wintel or Apple APIs.

Google News, introduced in 2002, is a web application that gathers news articles in real time, organizes them by topic and user preferences (inferred from tracking data) and provides readers with links to the original article. It offers a rich and interactive user interface displaying photos, videos and interactive frames as well as text. In contrast, Apple News is another news aggregator that runs "native" on Apple mobile devices and Macintosh desktops. Today Google News and Apple News are virtually indistinguishable in terms of functionality, even though they reach users via different routes—one as a website viewed through a browser, the other as an application downloaded onto the user's device.

The rich appearance, complex interaction, and fast response of Google News was made possible by a technology called "Ajax."[40] Classical web interaction took the form of:

Request—remote processing—load page
Request—remote processing—load page
Request …

Each new interaction required both remote processing and a full page reload. Each page took time to load while the user simply watched. In contrast, Ajax used pre-existing web protocols and platform independent languages to achieve dynamic partial page loads. The Ajax "engine"

---

in the next chapter.

[37] McKinnon and Glazer (2019).

[38] Statt and Brandom (2020).

[39] Zakrzewiski (2021).

[40] Ajax stands for "Asynchronous Java Script + XML." The languages originally used were XHTML and Javascript. The protocols initially used were CSS, the Document Object Model, XML and XSLT, and XMLHttpRequest. The name was coined by Jesse James Garret who was among the first to realize the significance of the new technology. Garrett (2005).

decoupled changes in what the user sees from communication with the remote server. Thus the server could do its work as other parts of the page were changing on the user's screen. Users' waiting time declined dramatically.

Ajax thus solved a critical technical bottleneck in the core flow processes that supported web browsing. Web applications could have the rich formatting and high levels of interaction that previously were possible only for "native" applications running on a local microprocessor and operating system. And because it was a combination of public protocols laced together in a fairly simple way, Ajax was not proprietary, hence could not be the basis for a new strategic bottleneck. Any developer could use the technology without splitting the value created with an owner.

Not surprisingly, Google was an early adopter of Ajax technology. Gmail, Google's email application, was the first web application targeted at the mass market to use it.[41] Google went on to use Ajax in Google News, Maps, Suggest, and Groups.[42] In 2020, Ajax and its descendants were employed across the web to deliver a wide variety of highly interactive web pages and applications.[43]

Web applications, even those using Ajax, do not guarantee platform independence, however. On mobile devices, applications developed for a particular device ("native" apps) are generally faster and more efficient in using the devices' resources (power, storage, CPU) and features (camera, accelerometer, etc.).[44]

In fact, the tradeoff for mobile app developers in 2020-2021 is identical to that faced by the developers of early PC applications and browsers. Recall that Lotus 1-2-3, written in the Intel assembly language, was much faster than Visicalc, which was ported from the Apple II. Because of its speed (caused by efficient core processes), Lotus quickly became the market leader in spreadsheets.[45] Two decades later, Netscape tried to develop a platform-independent browser, but had to use platform-specific instructions (Windows APIs and Intel instructions) to achieve acceptably fast page loads.[46]

21.9 Platform Envelopment—Apple iOS and Apps Store

Disintermediation via envelopment occurs when a platform that is already unique and essential, i.e., a strategic bottleneck, incorporates the functions of a subsidiary, stand-alone platform into its core processes and offers a combined product at a lower price than the sum of the prices of the two separate platforms.[47] The stand-alone platform cannot cut its price enough to

---

[41] Calore (2014).

[42] Garrett (2005)..

[43] https://www.quora.com/Do-programmers-still-use-AJAX (viewed 1/16/20).

[44] Horbenko (undated).

[45] See Chapter 15.

[46] Cusumano and Yoffie (1998). See Chapter 17.

[47] In some cases, the enveloper withdraws its original product from the market; in other cases users can still

be competitive with the combined platform without severely diminishing its own revenue.[48]

In some cases, the enveloper withdraws its original product from the market; in others users can still buy the original platform separately. In Chapter 17, we saw that Intel partially enveloped the chipset and motherboard with its microprocessor (customers could still purchase microprocessors without chipsets). Microsoft fully enveloped its Internet Explorer browser and partially enveloped the Office productivity suite into the Windows operating system.

Full or partial envelopment is a strategy often directed at subsidiary platforms that threaten to become strategic bottlenecks co-equal with the original platform. In the mid-1990s, online sales posed such a threat to the sponsors of standards-based platforms in the PC industry.

As described in Chapter 19, before the Internet, consumers purchased computers and peripheral devices in physical stores or via mail order companies. Software was sold in shrink-wrapped packages through stores and mail order catalogs. These physical channels were extremely inefficient with long time delays and inventory scattered at many locations. The arrival and diffusion of the commercial Internet made it feasible and efficient to sell hardware and software online. This in turn created large cost savings in transportation and logistics, which the transaction platforms could share with their suppliers and customers.

The threat to standards-based platforms lay in the fact that, if a single distributor, such as Amazon, became the sole route to market for their complementors, the distributor would become a strategic bottleneck (unique, essential, owned) in its own right. Sponsors of standards-based platforms thus had strong defensive reasons to fear concentration of their complementors' distribution channels. Creating an inhouse transaction platform for complements diminished this threat.[49]

The first envelopments of transaction platforms by standards-based platforms involved mobile devices. In 2001, Apple introduced a mobile audio player, the iPod. Two years later, it launched the iTunes Music Store. Apple went on to launch the iPhone in June 2007. The original phone could not run third-party applications.[50] Steve Jobs strongly opposed letting third-party complementors write applications for the device.

Notwithstanding Job's objections, hackers released a third party game app only a few weeks after the phone was introduced.[51] At the behest of Apple's board of directors, Jobs withdrew his opposition.[52] The company released a software development kit (SDK) supporting third-party apps in February 2008. Third party developers were now *encouraged* to write

choose to buy the original platform on a stand-alone basis.

[48] Eisenmann, Parker, Van Alstyne (2011).

[49] Interestingly, Microsoft and Intel, co-sponsors of the Wintel platform in the mid-1990s, were slow to recognize how online transaction platforms could weaken their strategic bottleneck in PCs.

[50] German and Bell (2007).

[51] Topolsky (2007).

[52] Reisinger, D. (2007); Isaacson (2011).

applications for the new phone.

In June 2008, Apple introduced the iPhone 3G with a full-fledged Apps Store and iTunes Wifi Store.[53] Hundreds of apps and thousands of songs were now downloadable directly to the device. The transaction platforms were wholly integrated into the iPhone's operating system. As a result, *the stores were the only route for any third-party products onto the device*. They were designed from the beginning to be strategic bottlenecks—essential, unique and owned by Apple.

As in the case of Intel's integration of the microprocessor and chipset (see Chapter 17), the integration of an operating system and a transaction platform served multiple purposes. First, it prevented a retailer like Amazon from becoming a second strategic bottleneck in the overall system.

Second, it allowed Apple to make the core flow processes within each device more efficient. Terry Crowley described how controlling the pathway to the phone made it feasible for many different applications to run seamlessly as modules on a low-powered device with limited memory:

> [Because apps were only available through the Apps Store,] Apple fully owned the application acquisition, install, update and uninstall process so could guarantee its safety. …
>
> In early versions of iOS, only the foreground application was allowed to run, preventing background activity from eating battery, network or impacting foreground responsiveness. Unresponsive applications would be aggressively terminated.
>
> It was the combination of the hardware *and these software and policy approaches that allowed Apple to construct an ecosystem where the set of applications could explode while still maintaining control over the overall user experience*.[54]

Lastly, Apple's tight control of the interface between applications and operating system prevented fragmentation and made it much easier to write apps for iOS than for the more open Android operating system, which today is Apple's primary competitor.[55]

Notably, in designing the iPhone, Apple did not attempt to become a vertically integrated firm supplying all necessary functional components used in its products. Apple had been vertically integrated in the 1980s and early 1990s, but this strategy failed in competition with the more open Wintel platform-and-ecosystem. (See Chapter 15.)

Instead, Apple chose to bring *four platforms*, each with its own ecosystem, together under unified governance. First, the Apple-designed system-on-chip (SOC) was integrated with a

[53] German and Bell (2008). At the time, because of bandwidth limitations, downloading purchased music required a wifi connection.

[54] Crowley (2017), emphasis added.

[55] Horbenko (undated); The Manifest (2018).

proprietary operating system (iOS). The resulting *standards-based platform* locked-in developers and users via control of a single set of APIs. Apple then designed a *logistical platform* to coordinate the modular production network that produced both chips and handsets. Finally *two transaction platforms*, the Apps Store and the Itunes Store, fully controlled the distribution of software and music to Apple devices. (In addition, the handset provided access to two communication platforms that were not under Apple's control: the telephone system and the Internet.)

In contrast to its strategy in the 1980s and 1990s, Apple outsourced components that were not critical to its core processes. It delegated chip fabrication to Samsung and TSMC and hardware assembly to FoxConn and Pegatron. It encouraged third-parties to develop complementary applications using its APIs and music producers to supply music through iTunes. Finally it sold handsets through numerous distribution channels including its own stores, telephone companies such as AT&T and Verizon, online retailers such as Amazon, and "big box" retailers such as Best Buy.

Apple's decisions as to which activities to perform itself and which to outsource reflected a fine sense of balance between modular option value, the efficiencies of synchronized flow processes, and protection from disintermediation. In Chapter 17, we saw how Intel and Microsoft protected their strategic bottleneck from disintermediation by making their core flow processes more complex, interdependent and opaque. Apple's defensive strategy is similar, but instead of simply integrating instructions and APIs, it has also integrated different types of platforms within its boundaries.

Apple was not the only company to envelop a transaction platform within a standards-based platform. The year 2008 saw two other cases also involving mobile devices. First, in April, Amazon launched the Kindle e-book reader and Kindle store. E-books readable on the Kindle device were formatted using Amazon's proprietary standard AZW: with minor exceptions, books using other formats could not be read on a Kindle. However, the owner of a Kindle device could browse, purchase, and immediately read books on the device from almost anywhere in the U.S. Later, Amazon allowed Kindle books to be read on other devices, but the company still controls the standards for digital conversion and purchases of books from the Kindle Store.[56]

Then, in October 2008, T-Mobile introduced the first smartphone to be based on the open-source Android operating system. From the beginning, Android phones included the Android Market for third-party applications. In 2012, Google combined the Android Market, Google Music and the Google eBookstore into a new transaction platform called Google Play. As of 2021, Google Play is Google's sole transaction platform for digital content. Google's sponsorship of Android is the subject of the next chapter.

---

[56] To avoid Apple's 30% tax on "in app" purchases Amazon maintains strict separation between Kindle readers and the Kindle Store. It is difficult (though not impossible) to buy Kindle books from an iPhone or iPad. Yarrow (2012); Franco (2018).

21.10 Conclusion—How Technology Shapes Organizations

In capturing value, sponsors of digital exchange platforms face similar problems to sponsors of standards-based and logistical platforms, but with a different emphasis. Sponsors of standards-based platforms are primarily concerned with maintaining backward compatibility and preventing reverse engineering. Sponsors of logistical platforms are concerned with managing the flow of material goods and preventing suppliers from integrating forward. In contrast, sponsors of digital exchange platforms are primarily concerned with the efficiency and scalability of their critical algorithms.

Three algorithmic processes are central to the functioning of all digital exchange platforms: (1) search and ad placement; (2) dynamic pricing (of products and ads); and (3) data analysis and prediction. These processes occur in milliseconds hence the underlying algorithms are highly interdependent. The processes are also shot through with transient bottlenecks in code, data centers, and networks. Addressing these bottlenecks in a timely way calls for a single organization under unified governance, making use of direct authority and managerial hierarchies. Organizations with distributed governance are less likely to succeed in either the day-to-day management or the coding of these processes.

Machine learning, artificial intelligence and automated experimentation are three supplementary technologies that make the core algorithms more efficient by improving predictions from data. These technologies are supermodular complements: advances in one make the others more valuable. However, unlike the core processes, they do not require tight synchronization. In 2021, the development of these technologies was spread among many organizations, including universities, large corporations, government agencies, and technology startups. Partnerships, open source projects, acquisitions, and technology licensing are common ways of organizing the underlying activities. Distributed modular complementarity (DMC) holds in these settings.

Data on the behavior and identities of individuals online can be used to improve predictions made by the platform's core algorithms. Personal data tracking does appear to have a significant positive effect on users' purchases and the effectiveness of advertisements. As a result, despite concerns about privacy, platform sponsors have a vested interest in continuing to track users' online activities and in sharing (and selling) the data they gather. At this point, it is too early to tell how the inherent tensions between platform sponsors' predictive ability and users' privacy will be resolved. Several jurisdictions including the EU, China and California have enacted or are considering regulation.

The core algorithms of digital exchange platforms determine speed and efficiency, but do not lock users into the platform itself. Three mechanisms create high user switching costs: (1) control of visible instructions that become interwoven in code; (2) storage of users' data; and (3) "specific" inertia, that is, value placed by ecosystem participants on specific other participants (rather than generic attributes of the platform). High switching costs permit platforms to survive even with high costs and inferior core algorithms. Thus to increase switching costs, many digital exchange platforms have become standards-based platforms, promoting proprietary APIs. Many are also providing free or low-cost storage of digital files. Finally, social media and messaging platforms often promote exclusive member-to-member ties.

Disintermediation is a risk for any platform sponsor. Like product platforms, exchange platforms can be disintermediated by substitution, reverse engineering, platform-independent complements and envelopment.

The root cause of *substitution* is generally inferior core processes, especially when one generation of technology is superseded by another. Modular processes are easier to upgrade piecemeal (see Chapter 11), thus a modular technical architecture is a defense against substitution. However, modular processes generally do not deliver the same level of speed and synchronization as integrated processes. Thus in the design of their core algorithms, sponsors of digital exchange platforms may not have any choice except to integrate.

Conveniently, integrated systems are also more immune to *reverse engineering* than modular systems, because the cost of reverse engineering goes up exponentially with the complexity and interdependence of the target system. We thus have the conundrum that, while a modular architecture makes a platform *less* vulnerable to substitution, it makes it *more* vulnerable to reverse engineering. (Recall that in the 1980s, the highly modular IBM PC quickly overtook the more integrated Apple Macintosh—see Chapter 15.)

Because there is no single best approach, platform sponsors must make nuanced decisions as to what parts of their systems to integrate (and insource) and what parts to modularize (and outsource). For example, the Apple iPhone displays a very high degree of integration in the vertical stack (microprocessor, handset design, operating system) and the Apps Store, but a high degree of modularity in handset components, chip and handset manufacturing, distribution channels, and third-party applications. Android phones are even more modular and more diverse in their designs, but suffer from a higher degree of fragmentation, so that not all applications work on all handsets.

Platform-independent complements by definition have many routes by which to reach end users. For this reason, they weaken the power of platform sponsors to maintain strategic bottlenecks. The interactive Worldwide Web (Web 2.0) and high-speed Internet connections made it possible to transfer complex computation to remote servers, bypassing local microprocessors and operating systems. As a result, highly functional and cheap devices, such as Chromebooks, are now on the market.

Despite their impressive achievements, platform independent devices and applications have not freed digital exchange platforms entirely from their dependence on proprietary standards-based platforms. Many users are willing to pay a premium for computers and mobile devices with a broader range of "native" capabilities than a "cloud-only" machine can offer. To the extent users prefer to own more powerful machines based on proprietary standards, sponsors of platform-independent web apps must split the value created by their products with the sponsors of proprietary product platforms. This is the price of complementarity.

Platform *envelopment* is a tactic used by larger platforms to simultaneously improve the efficiency of their core processes, deter reverse engineering, and undercut platform-independent complements. The basic idea is to offer a combination of essential and optional functions at a lower all-in cost than if the user purchased them as separate modules. The end result of envelopment is that platforms tend to become larger, more complex, and more integrated over time. *Envelopment thus has the same effect on platforms that vertical integration had on modern corporations at the turn of the 20th century.* By combining many different platforms within their boundaries, platform sponsors inevitably became big businesses in terms of revenue and profits, if not direct employees. In mid 2021, eight of the ten most valuable companies worldwide were sponsors of digital platforms.[57]

The next chapter describes how Google, the sponsor of numerous open communication platforms as well as the dominant transaction platforms for ad placement, used an open source operating system, Android, to attract handset makers and application developers into new ecosystem for mobile devices. Google's sponsorship of Android serves as an introduction to Part 5, which looks at how open source software practices are shaping both open source communities and corporations.

[57] In May 2021, the companies were Apple, Microsoft, Amazon, Alphabet (Google), Facebook, Tencent, Alibaba, and Tesla. Rankings taken from https://companiesmarketcap.com/ (viewed 5/12/21).

References

Arthur, W. Brian (1989) "Competing Technologies, Increasing Returns, and Lock-in by Historical Events," *Economic Journal*, 99(397): 116-131.

BBC News (1998) "Business: Yahoo! is still the first portal call," (June 5, 1998) http://news.bbc.co.uk/2/hi/business/107667.stm (viewed 1/2/20).

Calore, M. (2014) "Ten Years On, Gmail has Transformed the Web as We Know It," *Wired* (4/1/14); https://www.wired.com/2014/04/gmail-ten/ (viewed 1/16/20).

Chesbrough, H. (2006). *Open business models: How to thrive in the new innovation landscap*e. Harvard Business Press.

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

Crowley, T. (2017) "Why Windows on Arm?" *Hackernoon* (December 12, 2017) https://hackernoon.com/why-windows-on-arm-6cd9426f0923 (viewed 12/1/18).

Cusumano, M. A. and D. B. Yoffie (1998) *Competing on Internet Time* (New York: Free Press)

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

Franco, J. (2018) "Why You Can't Buy Books from the Kindle app on iPhone or iPad in 2018," *Techspot* (April 5, 2018) https://www.techspot.com/article/1597-how-to-buy-kindle-book-iphone-ipad/ (viewed 2/18/20).

Garrett, J.J. (2005) "Ajax: A New Approach to Web Applications," *Adaptive Path* (February 18, 2005) https://web.archive.org/web/20150910072359/http://adaptivepath.org/ideas/ajax-new-approach-web-applications/ (viewed 2/18/20).

German, K. and Bell, D. (2007) "Apple iPhone Review" *C/NET*, https://www.cnet.com/reviews/apple-iphone-review/ (viewed 11/19/18).

German, K. and Bell, D. (2008) "Apple iPhone 3G Review" *C/NET*, https://www.cnet.com/reviews/apple-iphone-3g-review/ (viewed 11/19/18).

Goldberg, S., Johnson, G., & Shriver, S. (2019). Regulating Privacy Online: The Early Impact of the GDPR on European Web Traffic & E-Commerce Outcomes. Available at SSRN 3421731.

Goldfarb, A., & Tucker, C. E. (2011). Privacy regulation and online advertising. *Management Science*, 57(1), 57-71.

Gray, M. (1996) *Web Growth Summary,* https://stuff.mit.edu/people/mkgray/net/web-growth-summary.html (viewed 8/16/21).

Grove, Andrew S. (1996). *Only the Paranoid Survive*, New York: Doubleday.

Gu, G. Y. (2021) "Technology and Disintermediation in Online Marketplaces" Paper presented at the Platform Strategy Research Conference, Questrom School, Boston University, Boston, MA (July 2021) http://questromworld.bu.edu/platformstrategy/program/ (viewed 7/26/21)

Gu, G., & Zhu, F. (2021). Trust and disintermediation: Evidence from an online freelance marketplace. *Management Science*, *67*(2), 794-807

Hart, Oliver and John Moore (1990) "Property Rights and the Nature of the Firm," *Journal of Political Economy*, 98(6):1119-1158.

Hennessy, John L. and David A. Patterson (1990) *Computer Architecture: A Quantitative Approach*, San Mateo, CA: Morgan Kaufmann.

Horbenko, Y. (undated) "Mobile Development: Choosing Between Native, Web, and Cross-Platform Applications, *Steel Kiwi*, https://steelkiwi.com/blog/how-choose-correct-platform-mobile-app-development/ (viewed 2/18/20).

Iansiti, M., & Lakhani, K. R. (2020). *Competing in the age of AI: strategy and leadership when algorithms and networks run the world*. Harvard Business Press.

Isaac, M. (2019) "Zuckerberg Plans to Integrate WhatsApp, Instagram and Facebook Messenger" *The New York Times* (January 25, 2019) https://www.nytimes.com/2019/01/25/technology/facebook-instagram-whatsapp-messenger.html (viewed 2/6/20).

Isaacson, W. (2011) *Steve Jobs*, J.C. Lattès.

Jacobides, M. G., Brusoni, S., & Candelon, F. (2021). The evolutionary dynamics of the artificial intelligence ecosystem. *Strategy Science*.

Johnson, G. A., Shriver, S. K., & Du, S. (2020). Consumer privacy choice in online advertising: Who opts out and at what cost to industry? *Marketing Science*.

Kapoor, R., & Agarwal, S. (2017). Sustaining superior performance in business ecosystems: Evidence from application software developers in the iOS and Android smartphone ecosystems. *Organization Science*, *28*(3), 531-551.

Kohavi, R., & Thomke, S. H. (2017). The surprising power of online experiments. *Harvard Business Review* (September-October).

Levy, S., 2011. *In the plex: How Google thinks, works, and shapes our lives*. Simon and Schuster.

Lewis, P. H. (1993) "Business Technology: Not Much New for Prodigy," *The New York Times*, (November 10, 1993) https://www.nytimes.com/1993/11/10/business/business-technology-not-much-new-for-prodigy.html (viewed 2/19/20)

Malik, O. (2019) "An Exclusive Look Inside Apple's A13 Bionic Chip," *Wired* (September 19, 2019) https://www.wired.com/story/apple-a13-bionic-chip-iphone/ (viewed 2/18/20).

McKinnon, J.D. and Glazer, E. (2019) "FTC Weighs Seeking Injunction Against Facebook Over How Its Apps Interact, *The Wall Street Journal* (December 12, 2019) https://www.wsj.com/articles/ftc-weighs-seeking-injunction-against-facebook-over-how-its-apps-interact-11576178055 (viewed 8/16/21)

Miller, R. (2016) "How AWS came to be," *Tech Crunch* (July 2, 2016) https://techcrunch.com/2016/07/02/andy-jassys-brief-history-of-the-genesis-of-aws/

Mueller, Scott (2005), *Upgrading and Repairing PCs*, 17th ed., Indianapolis: Que Publishing.

Netcraft (2020) “January 2020 Web Server Survey” (January 21, 2020) https://news.netcraft.com/archives/category/web-server-survey/ (viewed 2/19/20).

Reisinger, D. (2007) “Steve Jobs confirms third-party iPhone apps,” *Techradar*, https://www.techradar.com/news/phone-and-communications/mobile-phones/portable-devices/mp3-players/mobile-computing/tablets/audio/portable-audio/steve-jobs-confirms-third-party-iphone-apps-154618 (viewed 11/19/18)

Robertson, A. (2019) “How would opening up Facebook change the internet?” *The Verge* (October 23, 2019) https://www.theverge.com/2019/10/23/20926792/facebook-access-act-interoperability-data-portability-warner-hawley-bill-explainer (viewed 2/18/20).

Smith, M. (2018) “The Instagram API is Changing: Here’s What It Means for You,” *Later* (April 8, 2018) https://later.com/blog/instagram-api/ (viewed 1/11/20)

Statt, N. and Brandom, R. (2020) "The FTC is suing Facebook to unwind its acquisitions of Instagram and WhatsApp," *The Verge* (December 9, 2020) https://www.theverge.com/2020/12/9/22158483/facebook-antitrust-lawsuit-anti-competition-behavior-attorneys-general (viewed 8/16/21).

Tang, D., Agarwal, A., O'Brien, D., & Meyer, M. (2010). Overlapping experiment infrastructure: More, better, faster experimentation. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 17-26.

The Manifest (2018) “Android vs iOS: Which Platform to Build Your App for First?” *Medium* (February 22, 2018) https://medium.com/@the_manifest/android-vs-ios-which-platform-to-build-your-app-for-first-22ea8996abe1 (viewed 2/18/20).

Sun, T., Yuan, Z. , Li, C., Zhang, K. (2019) "The Value of Personal Data in the Internet Commerce: A High-stake Field Experiment on Data Regulation Policy" *Management Science* (forthcoming).

Toh, M. (2019) “Verizon is putting shopping at the heart of its media business,” CNN Business (June 24, 2019) https://www.cnn.com/2019/06/24/media/verizon-media-yahoo-future/index.html (viewed 1/2/20).

Topolsky, J. (2007) “First third-party "game" app appears for iPhone,” *Engadget*, https://www.engadget.com/2007/08/06/first-third-party-game-app-appears-for-iphone/ (viewed 11/19/18).

West, J., & Bogers, M. (2014). Leveraging external sources of innovation: a review of research on open innovation. *Journal of product innovation management*, 31(4), 814-831.

Wu, T. (2010). *The master switch: The rise and fall of information empires*. Vintage.

Yarrow, J. (2012) “Amazon Has Found A Clever Way To Avoid Giving Apple A 30% Cut Of eBook Sales,” *Business Insider* (January 11, 2012) https://www.businessinsider.com/amazon-kindle-ipad-store-2012-1 (viewed 2/18/20)

Zakrzewiski, C. (2021) "FTC refiles antitrust case against Facebook, says no other social network comes close to it scale," *The Washington Post* (August 19, 2021) https://www.washingtonpost.com/technology/2021/08/19/ftc-refiles-facebook-lawsuit-lina-khan/ (viewed 8/20/21).

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

Zuboff, S. (2019) *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power* (PublicAffairs).