

The Rise of Platform Ecosystems

1

May you live in interesting times.
Ancient Chinese curse

IN THIS CHAPTER

- What platforms are and what they are not
- Core components of platform ecosystems
- Drivers of the migration toward platforms in diverse industries

1.1 THE WAR OF ECOSYSTEMS

Blackberry had everything going right. It had fanatically loyal customers and its products were innovative, well engineered, durable, and got raving reviews from critics. After years of commanding a lion's share (about 50%) of the smartphone market that it largely created, it had trouble breaking past a 1% market share with its newest products by 2012, leading to its subsequent downfall. Blackberry assumed that the problem was Apple and then Google—both industry outsiders—who had since entered the fray. So, it did what made sense: Price more competitively, invest more in developing new products, upgrade its operating system, and step up marketing. Nothing worked. Its error was failing to realize that the basis for competition had changed: It was no longer Blackberry against Apple smartphones. Instead, it was the Blackberry ecosystem against the iOS ecosystem. It was not one product against another but Blackberry's army of 8000 external innovators against Apple's 200,000. Blackberry's mistake was failing to realize the ecosystem on which its continued success depended. All three companies made good products, but the lack of enough innovative apps muted Blackberry's market potential. It was already too late to catch up by the time Blackberry realized that the competitive blueprint had shifted. The *Red Queen* effect¹—the need to run faster just to stay in the same place—had taken over.

Platforms are creating an entirely new blueprint for competition—one that puts ecosystems in head-to-head competition. The ongoing migration from product and service competition to platform-based competition in many industries and markets is driven by forces—packetization of products, services, and activities; software embedding and ubiquitous networking of everyday objects; and the increased need for specialization—that are increasingly infusing characteristics of the software industry into

☆“To view the full reference list for the book, click [here](#) or see page 283.”

¹As the next chapter explains, one competitor successfully adopting a platform-centric approach can compress an entire industry's innovation clockspeed, requiring much greater effort just to stay in the same place.

many nontechnology industries. What served firms well in product-based markets can become their Achilles heel in platform-based markets. Managing platform-based businesses requires an entirely different mindset for strategy. Most of the old rules of business are alive and well, but many of the assumptions behind them do not hold in such environments. However, few observers have moved beyond individual superstar anecdotes of Apple, Facebook, and Amazon to analyze the broader principles and mechanisms that generalize beyond them. How should platforms be designed? How should they be governed, controlled, and priced? How can the work of so many be coordinated in the absence of familiar organizational structures? How do these choices shape their evolvability, their competitive durability, and their survival? How can their design create win–win propositions for app developers and users? How does a product or service even become a platform? The objective of this book is to provide you with actionable tools to arrive at your own answers about platform design strategies, and for platforms beyond the idiosyncrasies of the few poster children of the popular press.

This book is based on two premises. First, that the migration of competition from products to platforms—in technology and nontechnology industries alike—requires a different mindset for managing them. Second, evolvability in unforeseeable ways is key to thriving in platform markets but is rarely the dominant emphasis in complex software systems. Architecture and strategy are the two gears of a platform’s evolutionary motor that must interlock and align. Evolution² is therefore predicated in the *interplay* between its irreversible architecture and how it is governed. Platforms that thrive are ones whose *ecosystems* outpace rival ones in the evolutionary race. They orchestrate their ecosystems to leverage the drive and expertise of many outsiders without compromising ecosystem-wide integration. But not all platforms are created equal; the seeds of effective orchestration are sown in their early architecture. This is also where the landmines that lead to their collapse are hidden.

This chapter lays a foundation for introducing these ideas. It begins with an overview of software platforms and how they are changing the rules of competition. It then identifies the central elements of platform ecosystems and their competitive environment. It also clarifies what is *not* a platform. It then describes the five drivers of the migration toward software platforms in a variety of technology and nontechnology industries ranging from smartphones, appliances, fast food, craft machines, automobiles, medicine, and professional services. These include (1) the need for deepening specialization as firms struggle to deliver increasingly complex products and services; (2) the “packetization” of products, services, business processes, and activities; (3) the baking of routine business activities into software across a plethora of industries; (4) the emergence of the Internet of Things; and (5) the growing ubiquity of cheap, fast, and untethered digital networks. The *confluence* of these drivers can infuse properties of software platforms into products and services in mundane activities in nontechnology industries, making this book’s content surprisingly industry agnostic. This transformation can change how firms make money, retain customers, organize, and survive.

1.2 PLATFORM ECOSYSTEMS

Platform-based software ecosystems such as the iOS and its 800,000 “apps” produced by 200,000 firms or Facebook and its 9 million apps are increasingly becoming the dominant model for the software

²Evolution is multifaceted, as described in [Part III](#), encompassing ideas such as durability of competitive advantage, stickiness, enveloping adjacent market, innovation at the app and platform level, and creating derivative and “nested” platforms.

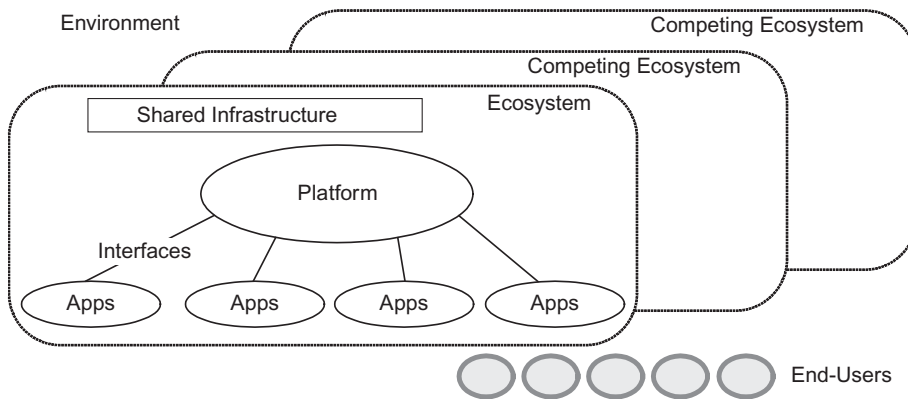
industry and digital services.³ The utility of almost any platform is increasingly shaped by the ecosystem that surrounds it. Take Apple's record-breaking iOS platform that includes the iPhone, iPod, and iPad. Its value to its 365 million users comes largely from the 800,000 complementary apps over which Apple has little ownership. Unlike traditional software development, platforms are designed to leverage the expertise of a diverse developer community—with ingenuity, hunger, skills, and an appreciation of user needs that platform owners might not possess. The emergence of such platform ecosystems (simply, systems composed of diverse smaller systems) is relocating the locus of innovation from the firm to a massive network of outside firms. The goal is to rapidly develop new capabilities and foster innovations unforeseeable by the platform's original designers. The idea of a platform as a foundation on which one builds is not new. Product families have existed in the tooling industry for decades, automotive platforms underpinned General Motors' dominance over Ford in the 1920s, operating systems are the engines in the IT industry, and “two-sided markets” that bring together buyers and sellers have existed since medieval times (Eisenmann et al., 2006; Fichman, 2004; Katz and Shapiro, 1994). But software infuses unique but poorly understood properties into platforms.

Our focus in this book is on *software-based platforms*, which create distinctively more complex opportunities and challenges than other types of platforms. They also function on a scale that is unprecedented in the industrial age because they allow literally hundreds of thousands of small companies to collectively do things that a traditional network of partners or intricate supply chains could not even dream of accomplishing. The potential power of platform ecosystems comes from leveraging the unique expertise of many, diverse independent app developers driven by market incentives on a scale that is impossible to replicate within a single organization. The platform model essentially outsources to thousands of outside partners innovation that used to be done inhouse, who bear all the cost and risk of innovating and then share the proceeds with the platform owner. The platform model throws the brainpower of thousands of small firms, mixes it with their hunger to succeed in the market, and lets the market determine the winners and losers. This potent mix of specialized expertise with the disciplining power of markets can foster innovation at a rate that exceeds by orders of magnitude conventional business models. Products that became platforms from 1990 until 2004 enjoyed a 500% increase in innovation, most of which came from outside developers (Boudreau, 2010). A platform's success therefore depends not only on the platform owner, but also on a multitude of ecosystem partners' ability to deliver (Adner, 2012, p. 1).

1.2.1 Elements of a software platform ecosystem

A platform-based ecosystem consists of two major elements—a platform and complementary apps—as Figure 1.1 illustrates. A software *platform* is a software-based product or service that serves as a foundation on which outside parties can build complementary products or services. A software platform is therefore an extensible software-based system that provides the core functionality shared by “apps” that interoperate with it, and the interfaces through which they interoperate (Baldwin and Woodard, 2009; Tiwana et al., 2010). We refer to the lead firm primarily responsible for the platform as the *platform owner*, sometimes also called the ecosystem's keystone firm (Iansiti and Levien, 2004) or the economic catalyst (Evans and Schmalensee, 2007). Platform ownership can be shared by multiple firms and a platform need not be proprietary or for-profit. An *app* refers to an add-on software

³Sources: <http://148apps.biz/app-store-metrics/> and <http://newsroom.fb.com/Platform>.

**FIGURE 1.1**

Elements of a platform ecosystem.

subsystem or software service that connects to the platform to extend its functionality. Although such complementary subsystems are often also called add-ons, plug-ins, modules, and extensions, here we refer to such platform complements simply as apps and their developers as app developers. Apps are complementary goods for platforms; platforms are functionally more desirable when there are a wide variety of complements available to them. (Two products are complements when one increases the attractiveness of the other; think of cookies and milk or a laptop and a Web browser.) For example, Internet streaming boxes are more desirable when streaming content is widely available; smartphones are more valuable when networks supporting them exist; Amazon's Kindle is more valuable when publishers produce e-books. The platform therefore consists of the enabling core technologies and shared infrastructure that apps can leverage. Apps access and build on the functionality of the platform through a set of *interfaces* that allow them to communicate, interact, and interoperate with the platform. The metaphor that science fiction fans can relate to is that the platform is like the *Starship Enterprise* and apps are like the little shuttlecrafts that dock into its myriad ports. The collection of the platform and apps that interoperate with it represents the platform's *ecosystem*. A platform ecosystem therefore meets the criteria for defining a complex system; one comprised of numerous interacting subsystems (Simon, 1962). Table 1.1 summarizes these core elements of a platform ecosystem.

Outside of these central elements of a platform ecosystem are three other contextual features: end-users, rival platform ecosystems, and the competitive environment in which they exist. End-users are the collection of existing and prospective adopters of the platform. The characteristics and diversity of this market evolves over time and as industries converge and split. A platform ecosystem exists within a larger competitive environment, often competing with other rival platform ecosystems. Such rival platform ecosystems constantly compete for both users and app developers. For example, Apple's iOS competes with Google's Android, BlackBerry, Nokia's Symbian, and Microsoft's mobile platforms. The competition within this environment is rarely directly among the platforms themselves but rather among competing ecosystems. The more intense this competition, the more important a platform's evolution becomes for surviving and thriving. A vibrant and dynamic ecosystem is therefore key to the survival of any software platform, and increasingly of products and services as they morph into platforms or become subservient complements of another platform.

Table 1.1 Core Elements of a Platform Ecosystem

Element	Definition	Example
Platform	The extensible codebase of a software-based system that provides core functionality shared by apps that interoperate with it, and the interfaces through which they interoperate	iOS, Android Dropbox, Twitter AWS Firefox, Chrome
App	An add-on software subsystem or service that connects to the platform to add functionality to it. Also referred to as a module, extension, plug-in, or add-on	Apps Apps Apps Extensions
Ecosystem	The collection of the platform and the apps specific to it	
Interfaces	Specifications that describe how the platform and apps interact and exchange information	APIs Protocols
Architecture	A conceptual blueprint that describes how the ecosystem is partitioned into a relatively stable platform and a complementary set of apps that are encouraged to vary, and the design rules binding on both	–

An ecosystem can also be divided into its upstream and downstream parts of a value chain, as illustrated in [Figure 1.2](#). The *upstream* part of the value chain is what goes into producing the platform itself (component and hardware suppliers, software licensors, manufacturing partners, network connectivity providers). The *downstream* part of the value chain includes platform complement producers (primarily app developers and complementary service providers), end-users who adopt it, and other intermediaries between the platform owner and end-users such as retailers and carriers ([Adner and Kapoor, 2010](#)). Apps are therefore *downstream complements* to a platform. [Table 1.2](#) provides examples of various contemporary platform ecosystems and their downstream complements. Downstream complements are bundled by a platform's end-users to customize the platform to their unique needs ([Adner and Kapoor, 2010](#)). Other downstream complements are necessary but insufficient to sustain differentiation of a platform vis-à-vis rival platforms. The platform itself therefore serves as only one part of the larger bundled system from which the platform's end-users derive value. The attractiveness of a platform to end-users comes not from the platform itself but from what they can do with it. The fate and survival of a platform then critically hinges on the diversity and vibrancy of its downstream ecosystem. The evolutionary battles of platform dominance and survival are fought primarily downstream, where formidable competitive barriers for rival platforms can be created. That does not mean that the upstream is unimportant; it just does not differentiate platforms in their evolutionary trajectory to the same degree as long as they manage the upstream task of executing the platform assembly process comparably efficiently. Our focus in this book is therefore exclusively on the downstream part of the platform value chain.

1.2.2 What a platform is not

We must also draw a sharp boundary of what a platform is *not*, particularly given the colloquial use of the label in ways that can mean many different things to different people. The common denominator of all platforms is that they facilitate interactions between two *distinct* groups (the two

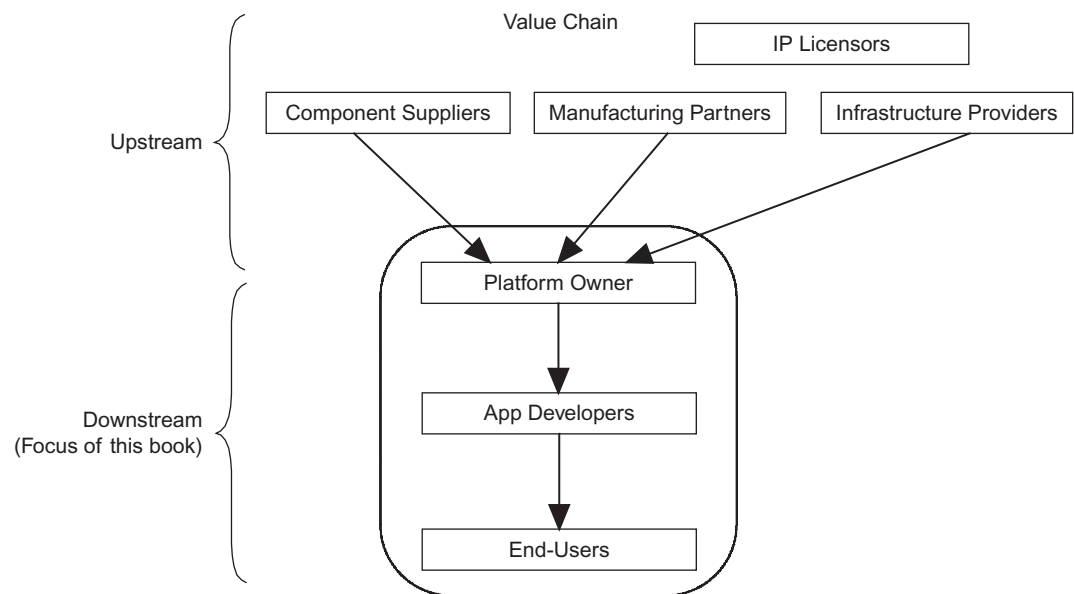


FIGURE 1.2
Upstream and downstream parts of platform value chains.

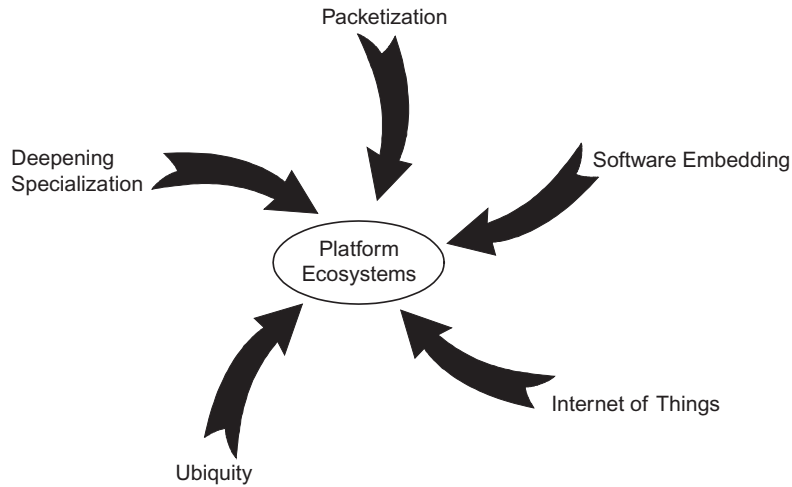
Table 1.2 Examples of Software Platforms		
Industry	Platform	“Apps”
Mobile computing	Blackberry OS	Apps
	Apple’s iOS	
	Google’s Android	
	Palm OS/HP TouchOS (defunct)	
Browser	Firefox	Extensions
	Google Chrome	Add-ins, Web apps
Social networks	Dropbox	Apps
	Twitter	
	Facebook	
Publishing	Amazon’s Kindle	e-Books
	iTunes	Music
Specialized software tools	R (statistical analysis platform)	Modules
Operating systems	Ubuntu Linux	Applications
Search engines	Users	Advertisers
Video games	Gamers	Game developers

“sides”) that want to interact with and need each other (Evans and Schmalensee, 2007, p. 38). The platform’s value to a user depends on the number of adopters on the other side. Our focus here is on multisided platforms rather than one-sided platforms, which we do not consider true platforms at all. Instead, they are products or services often confused or mislabeled as platforms. A platform by definition is at least two-sided (Eisenmann et al., 2006). Almost every example of successful platforms touted in the press—iPhone, Windows, Facebook, Skype, Amazon, eBay, Google, Firefox, and Dropbox—started out not as platforms but as standalone products or services that were valuable to end-users. Only after end-users widely adopted these products and services did they add a second side—developers—and transform into a platform. In one-sided platforms, the platform owner does not directly interact with two groups that might want to interact with each other; rather, it interacts primarily with one. Oracle, for example, provides enterprise systems to firms such as Target and Wal-Mart to enhance their business processes; however, it does not directly interact with their customers. Apple’s iOS, Google’s Chrome and Android, Mozilla’s Firefox, Ubuntu, Dropbox, Twitter, and Amazon Web Services (AWS) are examples of platforms. Platforms are not more complex versions of supply chains either. Supply chains are often one-sided markets rather than multisided markets, and therefore obey a different, simpler, and more predictable set of principles. We focus in this book on platforms in competitive consumer markets and exclude internal IT “platforms” that organizations build primarily for their own use. We also focus less on trading platforms that solely exist to match buyers and sellers of commodities (e.g., eBay and Amazon) in favor of software-based platforms where complementors actually contribute to the functionality and capabilities of the platform.

1.3 DRIVERS OF THE MIGRATION TOWARD PLATFORMS

Recent advances in technology are increasingly making it possible to reconfigure traditional industries along the lines of software-centric platforms abundant in ecosystem-creation opportunities. Much of this shift is facilitated by five drivers. These drivers, accelerating the migration from product and service competition to platform-based competition in a variety of diverse industries, are summarized in Figure 1.3 and their consequences in Table 1.3. These drivers include (1) deepening specialization within industries; (2) the “packetization” of products, services, business processes, and activities; (3) the baking of routine business activities into software; (4) the emergence of the “Internet of Things”; and (5) the growing ubiquity of mobile Internet protocol-based data networks. It is the confluence of these drivers that is transforming platforms into the de facto engines of new economic activity.

The graveyard of fallen giants is littered with once-dominant companies that failed to recognize these shifts in their own industry. They can make proven business models obsolete, alter the foundational assumptions of the industry, and require a different mindset to compete in an industry. It is not just the technology industries but any information-intensive industry that is likely to be affected by these drivers. And, as we explain in the subsequent chapters, some of these drivers can transform a low-skill, non-information-intensive industry into an information- and skill-intensive industry that then begins to behave like any other platform-centric industry. These drivers become even more forceful when they coexist, and their joint effects often exceed the sum of their parts.

**FIGURE 1.3**

The five drivers of the migration toward platform-centric business models.

Table 1.3 Consequences of the Five Drivers Toward Platform-Centric Business Models		
Driver	Description	Consequences
Deepening specialization	Increased need for deep expertise due to growing complexity of products and services	<ul style="list-style-type: none"> • Simultaneously shrinking and expanding firm boundaries • Red Queen effect from clockspeed compression • Increased interdependence among firms
Packetization	Digitization of “something”—an activity, a process, a product, or a service—that was previously not digitized	<ul style="list-style-type: none"> • Location-independent distribution ability of work • Deepening specialization
Software embedding	Baking a routine business activity into software	<ul style="list-style-type: none"> • Products-to-services transformation • Morphing physical–digital boundary • Convergence of adjacent industries
Internet of Things	Everyday objects inexpensively gaining the ability to directly talk using an Internet protocol	<ul style="list-style-type: none"> • Deluge of data streams from networked objects • Context awareness
Ubiquity	The growing omnipresence of cheap and fast wireless Internet data networks	<ul style="list-style-type: none"> • Loosely coupled networks rival efficiencies of firms • Alters who can participate from where • Alters where services can be delivered • Scale without ownership

1.3.1 Driver #1: Deepening specialization

Customers are increasingly demanding more customization instead of homogenous products and services delivered in volume (Williamson and De Meyer, 2012). At the same time, the complexity of products and services across diverse industries is also increasing. In software products in particular, the number of lines of code is estimated to double every 2 years (Evans et al., 2006, p. 303). For every line of software code in a typical software product in 2010, extrapolating this pattern for the following 20 years, as shown in Figure 1.4, suggests a dramatic growth in complexity over time. As products and services grow in complexity, it is increasingly difficult for one company—no matter how large—to specialize simultaneously in all domains that go into producing it (Evans et al., 2006, p. 53; Williamson and De Meyer, 2012). As knowledge needed to deliver increasingly complex products and services becomes more dispersed across many firms and many markets, no single firm or small network of firms can innovate alone (Dougherty and Dunne, 2011; Williamson and De Meyer, 2012). This creates a greater pressure for companies to more deeply specialize in their core competence and leave the rest to capable partners. This has increasingly led to the disaggregation of firms into complex supply chain networks involving many partners, and now into even larger ecosystems of smaller firms that specialize narrowly and deeply. These outsiders can potentially bring a breadth of deep insights about specialized domains, different application markets, and geographies that one company would struggle to maintain inhouse. Manufacturing, IT services, financial services, engineering, and even medical services industries are beginning to see this trend.

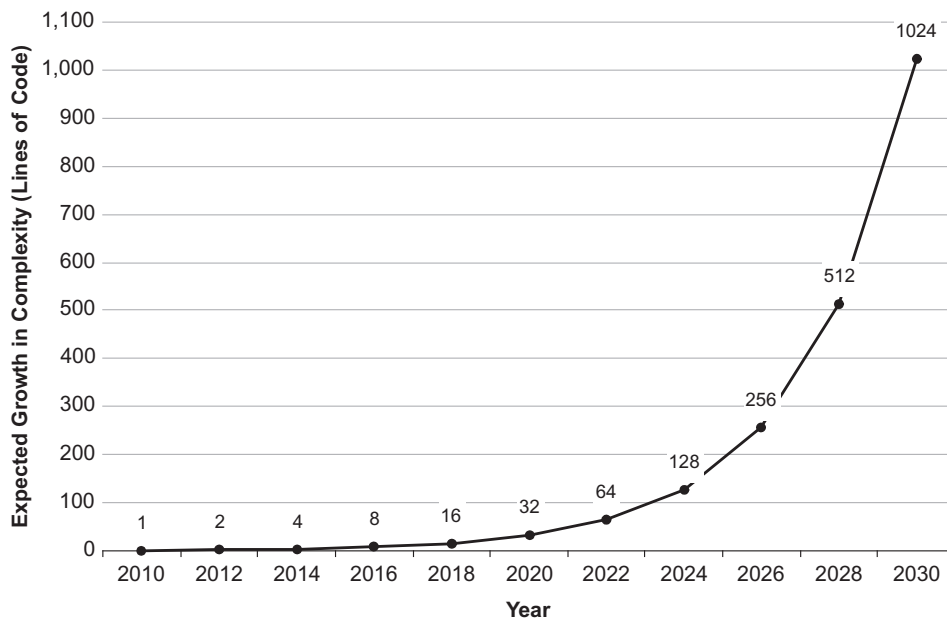


FIGURE 1.4

Extrapolating the increase in complexity for a single line of code in 2010 indicates a thousand-fold growth in complexity in two decades.

Location-dependence and coordination costs have often been the constraints that keep firms from disaggregating further, but the other four drivers—packetization, embedding of processes in software, the emergence of the Internet of Things, and ubiquity—are beginning to change that. A platform-centric approach enables pooling of multiple firms' knowledge bases that are more valuable in combination than in isolation.

1.3.1.1 Consequences

Deepening specialization has three consequences: (1) simultaneously shrinking and expanding firm boundaries, (2) the *Red Queen* effect, and (3) increased need for integration of distributed expertise. First, deeper specialization around their core competence means that companies are under pressure to do more of what they are really good at and less of everything else. The boundaries of the firm are therefore simultaneously contracting and expanding. They are contracting in the sense that the firm is focusing on a narrowing sliver of the value chain in [Figure 1.2](#)—the sliver where it focuses exclusively on its core capabilities. It is expanding in the sense that it relies on a broader community of outsiders to perform the complementary activities that go into delivering an attractive product or service. This is reducing the core activities that are done inhouse but simultaneously expanding the boundaries of the firm to include a greater assortment of outside partners. Ecosystems are an approach for dealing with such growing complexity without losing focus on what a firm does well. Platform-based businesses can potentially leverage the diversity and deep domain knowledge of a large pool of outside innovators, who are driven by market incentives. If managed right, the sheer pursuit of self-interest by these outsiders can accelerate the company's innovation around its core specialty far beyond what a lone corporation can possibly accomplish.

Second, one competitor successfully adopting a platform-centric approach can compress the clock-speed of innovation in the entire industry. It puts intense pressure on other firms in that industry to continuously innovate to keep their products and services differentiated, or else face the march of commoditization or outright irrelevance in a competitive marketplace. Firms in that industry cannot afford to ignore it; if their rivals are deepening specialization, they must too because staying in the same place is likely to require even more effort (see the *Red Queen effect* in [Chapter 2](#)). For example, Apple's iPad progressed through four generations within two years after its introduction, while Blackberry's Playbook struggled with its first generation. It did not matter that Blackberry once dominated the smartphone industry; the bar to keep up had been raised dramatically. This problem is not unique to the technology industries, as the examples throughout the book illustrate.

Third, this increased interdependence—across domains, markets, and geographies—between the firms needs to be managed. They need to be able to coordinate and integrate their dispersed contributions to deliver a valuable offering in the market. The mechanisms of authority, command-and-control, and contracts that underpin the very success of conventional firms rarely exist in ecosystem markets. Much of what makes traditional organizations efficient is precisely what can stifle innovation in large-scale ecosystems comprised of many independent firms. Modern business strategy focuses on “the firm,” not ecosystems; it emphasizes planning rather than serendipitous emergence, equilibrium rather than disequilibrium ([Dougherty and Dunne, 2011](#)). Not understanding how to manage such interdependence can keep companies from taking advantage of the very real opportunities that platform-based business models can offer. This requires managers to go back to the drawing board and rethink how to manage this increased interdependence between the firm and its complementors.

1.3.2 Driver #2: Packetization

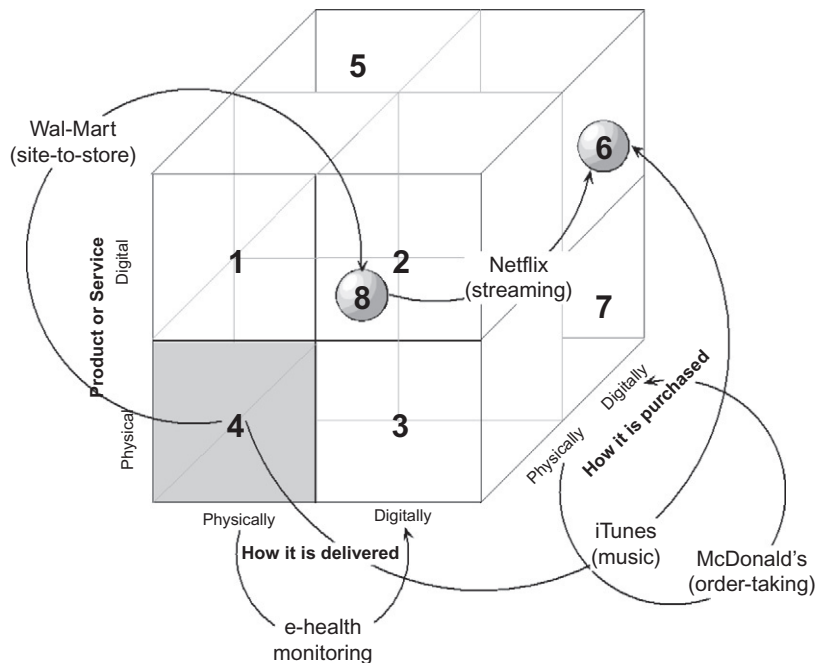
The second driver behind the rise of platforms is packetization. *Packetization* is the ability to digitize “something”—an activity, a process, a product, or a service—that was previously not digitized. Anything that can be digitized can be broken into Internet data “packets” and transported quite literally at the speed of light and at near zero cost across large distances. A digitized packet can be sent at nearly zero cost to someone a thousand miles away to work with, and then be sent back after processing all within fractions of a second. Digitization of music, books, and software is old news. The more interesting trends in digitization are in business activities that were previously thought as being location-dependent and localized.

Consider placing an order for a Big Mac in a McDonald’s drive-through window in your neighborhood. Taking those orders had to be done by an hourly employee who heard the customer through the drive-through microphone and punched in the order for the cooks in the kitchen, who then promptly delivered the Big Mac to the customer a minute or two later. Historically, this was considered a job impossible to outsource. It had to be done at the store by employees working right there. That was until McDonald’s figured out that a drive-through order was nothing more than a voice-based interaction—no different from a Skype conversation—and that the entire interaction could be packetized. This realization transformed McDonald’s order-taking processes. Today, when a customer pulls into the drive-through, rarely does she realize that the order taker on the other end of the speakerphone in Atlanta is several thousand miles away in Honolulu, Hawaii. The entire interaction occurs through voice packets that are digitized and sent to Hawaii, where a professional order taker enters the order into a computer that displays the output back at the customer’s store. The few seconds shaved from each order add up, the professionalization of order taking means a more consistent experience for customers and deeper specialization among McDonald’s employees. Packetization can therefore create specialization in even the most mundane activities that until recently could not be separated from their location.

Digitization need not occur on a large scale in a product or service industry to inject more platform-like attributes into it. Any product or service, or the business processes and activities that go into producing it, can be visualized along three dimensions that can be physical or digital: (1) the product or service itself, (2) how it is transacted or purchased, and (3) how it is delivered. A shift from physical to digital in even one of the three facets can open up platform opportunities. [Figure 1.5](#) illustrates this packetization cube framework with examples of digitization shifts in each of the three dimensions in a variety of industries such as retail, white goods manufacturing, entertainment, music, fast food, and healthcare. Think of your own industry to envision emerging opportunities; advances in technology are making it possible to accomplish packetization where it was not possible until recently. For example, additive manufacturing (also known as three-dimensional printing) is beginning to allow digital delivery of physical products. Additive manufacturing has the potential for transforming manufacturing just as the Internet transformed information, further blurring the boundary where the physical world ends and the digital realm begins.

1.3.2.1 Consequences

Packetization has two consequences: (1) it breaks the constraint of location-dependence and (2) it leads to deepening specialization. First, by digitizing even one aspect of an activity or process, it removes its location-dependence. The packetized activity can be completed anywhere that the packet can be sent. This opens up entirely new possibilities for division of labor—among humans and among machines—that goes into completing it. It can begin at the location where it originated, then its bits and pieces can

**FIGURE 1.5**

The packetization cube framework.

be distributed to workers in five different cities, and the completed output is delivered back at its point of origin. This means that even the most rote business activity can potentially be decomposed into pieces that can be executed in different places, by different parties, and then be reaggregated almost instantaneously. The Internet's near zero costs of communication and increasing speeds make it feasible to fundamentally rethink how the work to accomplish a packetized activity is partitioned and where it is done. Firms can now divvy up parts of the activity wherever the best cost/expertise combination is available in ways that were unimaginable until recently. Such packetization is creating a vast but largely invisible parallel economy that is on a trajectory to outgrow the physical economy in a couple of decades (Arthur, 2011). Business processes that once occurred among humans are now being executed among self-organizing machines, constantly leaving and reentering the physical world. Complexity theorist Brian Arthur describes this as the economy "growing a neural system" that is magnitudes of order faster, more predictable, and more productive than what it's replacing. By breaking free from geographical constraints, packetization of a product or service also vastly expands its potential market (Andreessen, 2011). Put another way, packetization is enabling entirely new business models that would have been the realm of fiction a decade ago.

Second, packetization is deepening specialization across all industries and professions. Part of the job of the jack-of-all-trades hourly employee at McDonald's has shifted to a specialist order taker, who just does more of the same activity. Similar trends of further decomposition of specialized activities are occurring across myriad professions including medicine, higher education, software development,

engineering design, accounting, financial services, customer support, and marketing. Consider a cardiologist in Richmond, Virginia, whose private practice sends packetized X-ray images to specialized technicians in Bangalore for interpretation and annotation overnight. The cardiologist can see more patients per hour because he is spending less time on lower value-added activities and more on his area of core competence. Therefore, professional work is becoming narrower and more specialized, and some of it is increasingly possible to delegate to machines that demand neither employee benefits nor breaks. This increased specialization and automation of rote work allows local specialists to focus on higher value-added tasks and on more creative work. The same consequence arises at the firm level as well. Packetization allows firms to increase their focus on even narrower aspects of their core competence where they can add most value, and potentially use a diverse ecosystem of specialist partner firms to provide the other pieces.

1.3.3 Driver #3: Software embedding

Software embedding refers to baking routine business processes or activity in software. Software systems are increasingly the invisible engine running many businesses (Evans and Schmalensee, 2007, p. 5). Routine business processes and day-to-day activities are becoming increasingly ingrained in networked software applications. Andreessen (2011), founder of Netscape, warns that software is “eating the world,” implying that it is taking over a growing percentage of the value chain of industries that are inhabitants of the physical world. Products and services across all types of industries have steadily growing software content, either directly in them or in their production value chain. Even a burger at your neighborhood McDonald’s. When you pay for a meal at a McDonald’s cash register, the order processing, payment processing, coupon validation, and kitchen notification is all embedded in software. Instead of a human performing the right sequence of tasks, the store’s software system manages it. An activity can therefore seamlessly move between the physical and digital realms, creating Brian Arthur’s (2011) so-called “parallel invisible economy.” Such software applications are increasingly networked, which means data across business processes can be aggregated instantaneously across stores, across firms, and across national boundaries. As such software embedding grows across product and service businesses in almost every industry, the performance of products and competitive delivery of services is increasingly dependent on software. As products and services become more software-intensive, software disproportionately shapes construction, delivery, and the end-user experience. Virtually every business is a software business to some degree, and software is increasingly gaining the power to make or break businesses and business models. Pricing and revenue models of traditional industries then begin to resemble the software industry, and an appreciation of platforms becomes increasingly vital to their sustained competitive advantage. The embedding of software is also transforming products into more platform-like services. For example, a major source of revenue for embroidery machines sold by Bernina and craft cutting machines by Silhouette is from selling digital patterns and designs that customers load onto the machines. Most of these digital designs are produced by independent third parties, mostly mom-and-pop design firms. This is effectively turning something as unexpected as a craft toolmaker into a platform owner. The perils of not fully appreciating the growing software-intensity of nonsoftware products are illustrated by Toyota’s 2009–2011 recall of 9 million cars that cost the company several billion dollars, attributable almost exclusively to software glitches. As the software content of almost every industry—automobiles, insurance, retail, manufacturing, aviation, even fast food—increases, the difference between them and a software company is blurring.

Although even the most complex products have largely been self-contained, they are now becoming a part of something larger (de Weck et al., 2011, p. 7), morphing into what can be described as “systems of systems” (de Weck et al., 2011, p. 13). At that point, a nontechnology company’s products and services inescapably acquire some properties of software platforms and become increasingly subject to their evolutionary dynamics.

1.3.3.1 Consequences

Software embedding has three consequences: (1) transformation of products into services, (2) morphing of the digital–physical boundary, and (3) convergence across industries. First, as a greater proportion of products are embedded in software, the opportunities to transform them into services increases. This trend is widely observed in the IT industry, where software such as mail clients have been replaced by services such as Gmail; licenses for databases have been replaced by utility-like storage services; and costly enterprise systems are increasingly being replaced by Web-based services. Kraft’s Tassimo coffee machines, for example, can report what is being brewed by its users and when, turning it into a platform—at least from Kraft’s perspective—for real-time intelligence gathering about consumer behavior and trends. In the automotive industry, cars increasingly offer subscription-based conveniences such as traffic monitoring, live navigation, and driving-based maintenance scheduling; this is an indication of their growing service content. The upside of this transformation is that services, unlike products, offer the potential for lock-in and a revenue stream as opposed to a one-time sale of a product. Firms that can make this transition can fundamentally alter their revenue model. However, the combination of the other four drivers also means that firms are increasingly going to be reliant on a vast network of specialized outside firms to be able to deliver on their own promises.

Second, increased embeddedness of business processes and activities in networked software further blurs the boundary between the digital and physical world. This requires firms in just about any nontechnology industry to take the intertwined software content of their own work as seriously as their bread-and-butter domain. Third, it simultaneously opens up opportunities and threats of convergence. Convergence means that two industries that were formerly independent are becoming mutually competitive and overlapping (Messerschmitt and Szyperski, 2003, p. 207). As two adjacent industries begin to bleed into each other, it creates opportunities for one to swallow the other by offering the incumbent firm’s product functionality or service as part of its own. (Also see envelopment in Chapter 2.) Platforms are powerful vehicles for converging markets. Apple, for example, has used its iOS platform to begin swallowing the adjacent online gaming console industry. Amazon has used its Kindle platform to become a book publisher and its AWS platform to begin swallowing segments of the IT services industry.

1.3.4 Driver #4: The internet of things

The fourth shaping force is the emerging Internet of Things, which allows everyday objects embedded with inexpensive sensors—pacemakers, shoes, tires, coffee machines, thermostats, billboards, and washers and dryers—to directly communicate using the same protocol that connects the Internet. The declining cost and increasing speed of semiconductors increasingly allow computing and communication capabilities to inexpensively be embedded in everyday objects. The number of devices connected to the Internet outgrew the number of humans in 2008; 50 billion “devices” will be connected to the Internet by 2020 (Jackson, 2011). Payments using mobile devices had crossed the \$1 trillion

threshold by the beginning of 2014. Smart meters that optimize electricity usage, vending machines that adjust prices based on weather, health monitors that report anomalies to physicians; CT scanners that beam diagnostic scan images; coffeemakers and washing machines that sense user behavior; and cows whose sensors transmit 200 megabytes of health data each year (Jackson, 2011; Thompson, 2010) all mark the beginnings of this shift. You might skeptically ask how such objects will be powered. Won't we start running out of IP addresses if everything gets networked? Such devices require little power to run, which can feasibly be extracted from thin air by tapping into ambient electromagnetic radiation (the cell phone signals and Wifi signals around you) (Economist, 2010b). And Internet addresses are not likely to run out with the adoption of an upgraded Internet protocol (IPv6), which allows 100 unique addresses for every atom on Earth (i.e., 340 trillion, trillion, trillion devices and objects can have their own Internet addresses). This book therefore refers to these artifacts as networked devices rather than machines or computers.

1.3.4.1 Consequences

The emergence of the Internet of Things has two consequences: (1) the generation of a deluge of data streams from networked objects and (2) increased context awareness. First, today's business models are based on largely static stocks of information. Businesses collect data on sales, clean it, analyze it, and use the analysis to inform business decisions. Some do it better than others, but the data is assumed to be a *stock* of data. The Internet of Things changes it to a constant deluge of data *streams* rather than data points that businesses are used to. An apt metaphor is a comparison between Niagara Falls and a pond; knowing how to swim in the latter won't get you very far in the former.

Second, networked objects can also communicate contextual information such as location, surroundings, and temperature in real time. The pathways of information are therefore changing, making objects in the physical world parts of an information system (Chui et al., 2010). Their ability to "talk" to other objects and systems in real time is making them smarter, autonomous, and more cooperative. Such contextual awareness makes entirely new business models and pricing strategies feasible. Whether it is tracking pallets in supply chains, monitoring driver behavior to determine insurance premiums, monitoring wear and tear on a car's tires to assess safety, or monitoring window-shopping behavior of potential customers in a store to determine optimal pricing that will lead to a sale, the commonality is the increased information intensity of doing business and the greater context awareness. This is increasingly fusing processes and objects in ways that make physical things parts of business processes. Turning these advances into a more dynamic, sense-and-respond capability demands new ways of handling the deluge of real-time data streams and even more specialized skills to turn it into a competitive advantage. It is almost impossible for one firm to do all of this, in turn requiring turning to ecosystem partners to contribute complementary capabilities. As some firms in an industry begin to exploit this emerging trend, it will further compress the timeframe within which other firms can exploit this data.

1.3.5 Driver #5: Ubiquity

The fifth driver of the rise of platforms is the omnipresence—or ubiquity—of increasingly cheaper (see Figure 1.6) and faster (see Figure 1.7) wireless data networks built around the Internet protocol. The ability to packetize something would be worthless if the packets were expensive to rapidly move over networks, or if the recipient had to be tethered to a desktop or physical location to receive them. The growing ubiquity—the property of being present everywhere—of Internet-based data networks

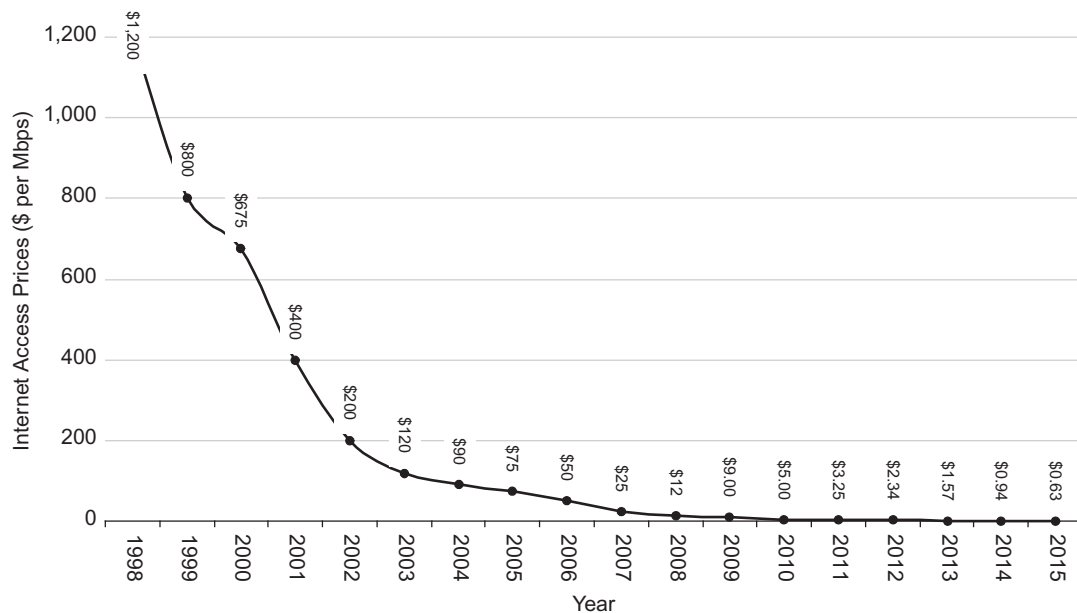


FIGURE 1.6

Internet access costs have been declining by a third every year.

Raw data source: <http://drpeering.net/white-papers/Internet-Transit-Pricing-Historical-And-Projected.php>.

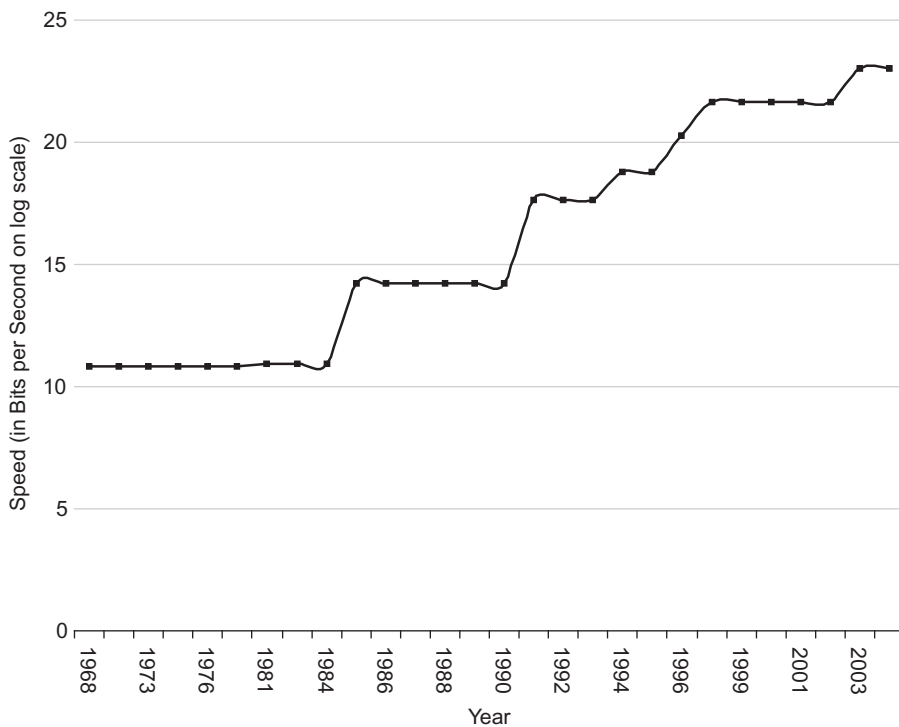


FIGURE 1.7

Internet backbone speeds increase by an order of magnitude every decade.

Raw data source: www.singularity.com/charts/page81.html.

is changing that. Furthermore, data communication costs are declining and network speeds have increased dramatically in the past decade. The speed of data networks on smartphones and mobile tablets far exceed those of land-based broadband connections a decade ago, and cost a fraction. Most of these networks all speak the same language—TCP/IP—the Internet protocol. This means that anything that can be packetized can potentially be delivered to end-users just about anywhere: at home, at work, in a store, in a car, in the air, or on a job site. Similarly, packetized activities can inexpensively be performed by geographically dispersed individuals. But, where will so much wireless bandwidth come from if trillions of things were to be networked? The freeing-up on the spectrum used for the now-defunct analog television broadcasts is helping create even longer-range networks capable of wirelessly moving Internet data at blazing speeds (400–800 mbps) ([Economist, 2010a](#)), further accelerating ubiquitous connectivity. The costs of entry of a new startup using digital infrastructure have declined a hundredfold in the last decade; a basic Internet application that used to cost \$150,000 a month 10 years ago now costs \$1500 ([Andreessen, 2011](#)). We are at a very nascent stage of truly ubiquitous connectivity, but we are progressing toward it in leaps and bounds.

1.3.5.1 Consequences

The primary consequence of ubiquity is that it will allow loosely coupled networks of small firms to rival the efficiencies of large firms. By banding together and helping coordinate distributed resources and participants in producing products and services, it turns the conventional model of scale through ownership to one of scale without ownership. Coupled with the trends toward packetization (driver #2), it alters who can participate in producing products and services and from where. It also alters where they can be delivered and from where. Location will matter less because companies can compete from anywhere, with rapidly eroding local advantages. Just as Kenmore can troubleshoot networked washing machines in an owners' home in Atlanta from a thousand miles away ([Prentice, 2010](#)), it can do the same 10,000 miles away in a customer's home in Japan or Australia. They also allow companies to crowd-source and "unsource" activities to their consumers, who help each other solve problems and do the work that was previously done by inhouse employees ([Economist, 2012a,b](#)). Furthermore, companies that were locationally disadvantaged can now compete without that handicap. Amazon, for example, has exploited ubiquity using smartphone apps that allow consumers to do price comparisons by scanning product barcodes in their competitors' stores. Using GPS data sent with every price scan, Amazon can crowd-source pricing intelligence-gathering to its own customers. Ubiquity allows specialization to coexist with collaboration; mass collaboration and crowd-sourcing become feasible and cost-effective; and expertise distributed across a company's locations no longer needs to remain in silos. A customer interaction can instantaneously and cost-effectively bring together specialized contributors dispersed across multiple continents. For example, American Express uses this shift to be able to seamlessly shuttle customer support calls between specialized agents in India and Florida in real time. However, exploiting growing ubiquity in sustainable ways that continue to competitively differentiate firms requires putting on a platform thinking cap.

1.3.6 The perfect storm

It is tempting but flawed to think that software-based platforms apply only to the IT industry. Platform markets are rapidly emerging in a variety of non-IT industries because these five drivers do not exist in isolation. It is the *combination* of these drivers that is creating unprecedented opportunities to transform

products and services across a wide variety of industries into software-centric platforms. The presence of one driver can make another driver more forceful, leading to a joint effect that exceeds the sum of their parts. Platform ideas are industry agonistic and the ecosystem dynamics that this book explores are applicable to a growing number of diverse industries. Innovation ecosystems like the ones on which this book focuses are increasingly emerging in the mortgage, finance, drug development, microprocessor, software, automotive, healthcare, banking, food services, and energy industries (Dougherty and Dunne, 2011). While it might be decades before this potential is fully realized in many industries, they will nevertheless begin exhibiting platform-like properties before then. It is therefore important for managers in just about every industry to understand the blueprints of platform-based markets to recognize and adapt to these shifts.

1.4 LESSONS LEARNED

Platforms are challenging conventional models that served us well in the industrial era. The confluences of five forces is creating the perfect storm fertile in opportunities to transform diverse industries into software-centric platforms. This transformation can alter how a firm makes money, retains customers, organizes everything from mundane tasks to innovation, and ensures its own survival. A brief summary of the core ideas described in this chapter appears below.

- *Platforms are changing the rules of competition.* Competition is migrating to rival platform ecosystems competing against one another, replacing competition among rival products and services. The potent mix of specialized expertise with the disciplining power of platform markets can foster innovation at a pace that can trump even the mightiest product and service business. Companies that fail to recognize this shift in their own industries are on a death march, on the waiting list of fallen giants like the Pony Express, Kodak, Blackberry, Palm, and Sony. The slower managers are in recognizing this, the more likely their companies are to become victims of the Red Queen effect, unable to catch up.
- *Platform ecosystems are composed of externally produced complements that augment the capabilities of the platform.* A software platform is an extensible software product or service that serves as a foundation on which independent outside parties can build complementary products or services (“apps”) that interoperate through the platform’s interfaces. The collection of the platform and apps that interoperate with it represents the platform’s ecosystem.
- *Innovations that differentiate platforms often emerge in the downstream part of the platform’s value chain.* A platform ecosystem’s value chain has upstream and downstream parts. The upstream part of the value chain is what goes into producing the platform itself. The downstream part includes app developers who produce capability-augmenting extensions to the platform. End-users can uniquely mix-and-match which of these downstream complements they include in their own instantiation of the platform ecosystem.
- *A platform without at least two distinct groups or “sides” is not a platform.* A true platform must be at least two-sided and span at least two distinct groups such as app developers and end-users that interact with each other using the platform. Platforms should not be confused with one-sided products and services, or with supply chains, all of which obey a different, simpler, and more predictable set of principles.

- *Most successful platforms began as standalone products or services.* Many successful platform examples touted in the press—iOS, Windows, Facebook, Skype, Amazon, eBay, Google, Firefox, Salesforce, and Dropbox—started out as standalone products or services that subsequently transformed into platforms by adding a second distinct group.
- *The confluence of five drivers is accelerating the migration toward platforms in a variety of industries.* These drivers include the need for deepening specialization within industries as they struggle to deliver increasingly complex products and services; “packetization” of products, services, business processes, and activities into digital bits; baking of routine business activities into software in industries as diverse as fast food, medicine, healthcare, automotive, education, engineering, accounting, financial services, and marketing; the emergence of the Internet of Things that allows everyday objects to communicate using the Internet; and finally the growing ubiquity of cheap, fast, and untethered Internet protocol-based networks.
- *Platform ecosystem ideas in this book are industry agnostic.* As the five drivers described in this book begin to affect various nontechnology industries, they infuse properties of software platforms into their products and services. Managers—independent of their industry—should therefore grasp the ideas described in this book so they are able to recognize and are ready to embrace platform opportunities and threats in their industries. As the old saying goes, fortune favors the prepared mind.

The next chapter introduces 10 core platform concepts that will provide us a shared vocabulary for the subsequent chapters in this book. These include the multifaceted notion of platform lifecycles and distinctive properties of software platforms such as network effects, multisidedness, different types of lock-ins, multihoming, tipping, and envelopment. It also explains nine guiding principles that we build on in later chapters on platform architecture, governance, and evolution. Some of these principles relate to getting platforms off the ground (e.g., the chicken-or-egg problem and the penguin problem), others relate to their design (e.g., the seesaw problem, the Humpty Dumpty problem, and the mirroring principle), and the last set guides their evolution (the Red Queen effect, emergence, the Goldilocks rule, and coevolution).