

Digital Service Platform Evolution: How Spotify Leveraged Boundary Resources to Become a Global Leader in Music Streaming

Daniel A. Skog
Umeå University
daniel.a.skog@umu.se

Henrik Wimelius
Umeå University
henrik.wimelius@umu.se

Johan Sandberg
Umeå University
johan.sandberg@umu.se

Abstract

Research on digital platform evolution is largely focused on how platform-owners leverage boundary resources to facilitate and control contributions from external developers to extend the functional diversity and scope of a digital device. However, our knowledge of the digital platforms that carve out their existence exclusively in the service layer of industry architectures, i.e. without proprietary device connections, is limited. The concept of digital service platforms directs attention to such platforms, the role of end-users as value co-creators, and devices as requisite, but not necessarily proprietary, distribution mechanisms for service. Based on a longitudinal case study of Spotify, this paper contributes by demonstrating that digital service platform evolution is characterized by specific architectural conditions that rationalize the use of boundary resources for extending scale rather than scope, and for resourcing and controlling not only developers but also end-users as a means to strategically adjust the evolutionary process.

1. Introduction

Digital platforms carry unprecedented possibilities for innovation and rapid growth. As demonstrated by firms such as Apple, Google and Facebook, digital platforms enable platform-owners to cultivate masses of external contributions from distributed and diverse actors, and ultimately to dominate whole industries. Therefore, it is perhaps not surprising that the concept of digital platforms has received increasing attention in Information Systems (IS) research [18].

Often based within contexts of mobile platforms [e.g. 2, 3, 4, 5, 8], the core empirical locus of research on digital platforms has historically been proprietary ensembles of devices, Operating Systems (OSs) and app stores. Building on a product oriented perspective on platforms as a means to achieve economies of scope [20], this stream of research has explored how

modularity can be exploited to enable functional diversification of a digital product [18, 24]. In resemblance to an ecosystem oriented view on platforms [20], diversification is achieved by making interfaces available outside of the manufacturing firm and the supply chain so as to enable the generation of complements from ecosystems [7].

Partly due to their layered architectures, digital platform dynamics is characterized by both unprecedented flexibility and hierarchical dependencies [14, 24]. Within such architectures, modules are largely product agnostic. This enables a layer of devices to draw upon a multitude of functionalities from a service layer. In turn, services can easily combine with other services, but also draw upon and combine diverse elements from a content layer [24]. Simultaneously, layered architectures are hierarchical [13] with dependencies between layers since digital content and services cannot be consumed without a digital device. Hence, based on its architectural position, the strategic rationale for how a digital product platform (defined here as a digital platform that includes a proprietary device) should leverage interfaces is clear. Through the design of boundary resources [8] a digital product platform can not only induce functional variety by mobilizing external developers, it can also become a centralized gatekeeper for content and service distribution from which it may profit.

However, analogous to both general platform theory [20] and service research [15, 16], little has been done within IS research to address, conceptualize and theorize service platforms. Digital service platforms that consist solely of service layer elements and operate on-top-of digital product platforms [18], has so far remained untheorized. Without a proprietary device, digital service platforms are likely to face both challenges and opportunities in relation to the flexibility and dependencies of layered architectures, yet our knowledge of how they can respond to or exploit them is severely limited.

Against this backdrop, we explore the following research question: What are the architectural characteristics that govern digital service platform

evolution and how are these leveraged to achieve strategic advantage? We explore this question through a longitudinal case study of Spotify - a digital platform for music streaming. Through process analysis, we trace its evolution from the official launch in 2008 until it was pronounced as the globally dominant service for digital music streaming in 2015.

2. Digital Platforms

Platforms have been explored from different perspectives in the literature, including product oriented and ecosystem oriented views. As the dominant perspective, the product oriented view is grounded in engineering design and in studies of the architectural structures of physical products that enable economies of scale and scope through the re-use of common assets and components. Through modular architectures with product-specific standards and interfaces, product platforms enable components to be efficiently re-used across product families, and new modules to be developed and added to extend the functionality of a given product [20]. Hence, the boundaries of a product platform are determined by the boundaries of a focal product and its manufacturer. In contrast, the ecosystem perspective portrays platforms as the hubs of wider business systems that facilitate, coordinate, and control exchanges between multiple organizations. Instead of being bounded to a single manufacturer or supply-chain [7], standards and interfaces are leveraged by platform ecosystems for external distribution of activities, enabling economies of scope and scale through mobilizing dispersed specialized actors [20].

Both the product and the ecosystem view is reflected in the concept of digital platforms emerging in IS research [18]. For example, a digital product platform has been conceptualized as an ensemble of proprietary elements that functions as a product in itself, but that simultaneously enables others to innovate upon it [24]. By supplying core assets and dedicated interfaces through which such assets may be retrieved, digital product platform-owners seek to attain economies of scope, *i.e.* to generate diversification of product functionality at relatively low-costs [18]. Analogous to the ecosystem perspective, this is achieved by making core assets and interfaces accessible outside of the manufacturing firm and the traditional supply-chain [7] to ecosystems of widely dispersed and heterogeneous developers [e.g. 3, 4, 8].

Digital platforms nevertheless also carry properties that distinguish them from other types of platforms [18]. First, in contrast to the product view, digital

platforms are based on and embedded within layered architectures [14, 24]. The architectural layers of digital products include a device layer where a focal device supplies the hardware and OS, a layer of service modules that provides applications that support end-users in their work with specific tasks, and a content layer provides *e.g.* digital audio, text and images [24]. The layered architecture enables unforeseen possibilities for extending the functionality of digital products after they have been manufactured. Through universal standards and interfaces, a digital device can host a wide range of services that combine a wide range of content. As a result, rather than imbued by design, the functionality of a digital product materializes at the time of use [24] as a particular device is utilized to access a certain service that draws on certain content (*e.g.* the use of Google Maps on a phone to get directions). As access to digital services and content necessitates the use of a digital device, the layered architecture is also characterized by hierarchical dependencies. These dependencies can be exploited with a digital product platform that erects boundaries around a device and thus provides control over service and content distribution through it. Second, the possibilities for economies of scale associated with product platforms is not directly translatable to digital product platforms because the marginal costs of reproducing a digital element is intrinsically low [10].

Hence, a focal concern for digital platform research is how digital product platforms can be organized so as to utilize the flexibility of a layered architecture to generate product diversity and simultaneously retain control over critical resources, avoid functionality drift, and ensure value-appropriation [21]. To that end, the concept of boundary resources has been utilized to understand how this paradoxical task can be managed [5]. Defined as *“the software tools and regulations that serve as the interface for the arm’s-length relationship between the platform owner and the application developer”* [8:176] boundary resources usually reflect the Application Programming Interfaces (APIs) or Software Development Kits (SDKs) with associated rules that platform-owners supply to external developers. By (re)designing boundary resources, a platform owner can both engage in resourcing, *i.e.* to facilitate the generation of inbound complements that extend the scope and diversity of the platform, and securing, *i.e.* to regulate the behaviors of contributors to avoid threats to the platform [8].

In summary, research on digital platforms has built on product and ecosystem perspectives and extended them by highlighting the strategic conditions that digitalization brings to the concept of platforms.

However, the platform concept suffers from a lack of empirical grounding in service contexts [20], and whereas there may be much to gain from translating platform theory and strategies to service research and practice, this remains a prioritized yet unresolved issue [15, 16]. So far, digital platform research has done little to remedy this situation with its dominant focus on digital product platforms as a means to cultivate contributions from external developers to extend the functional diversity of devices. In contrast, a digital service platform is comprised solely of service layer elements and is thereby void of a proprietary device. The concept of digital service platforms extends our attention from a sole focus on external developers to include the role of end-users as co-creators of value [14, 19] and the interfaces that makes this possible [1]. Moreover, it emphasizes the role of devices as distribution mechanisms for service [14], which manifests as a dependency to external digital product platforms for service distribution. From studies on the interaction between digital product platform owners and service layer contributors, we know that these dependencies may lead to power struggles [5]. However, empirical investigations into how such dynamics translate to digital service platform evolution is lacking in extant research.

3. Research design

We employed a longitudinal case study design with the aim of making analytical generalizations [23] to digital platform evolution theory. To that end, Spotify was chosen because it represents a successful case of digital platform evolution, and because it exhibits architectural conditions distinct from other frequently studied digital platforms. In contrast to mobile product platforms (e.g. Android or iOS), the Spotify platform solely occupies the service layer [24] and does not rely on a coupled proprietary device. It is therefore dependent on external devices for service delivery but its rationale for supplying interfaces to third-parties is independent of extending the scope of a particular device. Ultimately, Spotify categorizes with digital service platforms without a proprietary device that, as its core service, mediates service exchange between commercial content producers and end-customers, e.g. Netflix or Hulu.

3.1. Data collection

Digital platforms and ecosystems are intrinsically emergent, complex, and distributed, which suggests that understanding their evolution requires a diverse set of data [18]. Inspired by recent studies of digital

platform evolution [5, 8] we therefore relied on publically available data from several sources to amass a comprehensive empirical material as the basis for our analytical efforts.

First, we collected 1813 blog posts published in Spotify's 5 official blogs between 2006 and 2016. The blogs contained information targeting different actors, *i.e.* end-users, artists, developers, computer engineers, and the press. Second, 24 annual reviews from 6 different companies within the Spotify group provided insights into financial performance, perceived threats and opportunities, acquisitions and other key events. Third, we collected all results returned from the search query "spotify" on technology news blog aggregator Techmeme [5]. An additional 502 Spotify-related posts were thus retrieved from a wide selection of technology news providers. Supplementary data also helped to provide both a general understanding and specificity to our case. To that end, 12 yearly International Federation of the Phonographic Industry (IFPI) Digital Music Reports provided a deeper contextual understanding of the music industry prior to, and during, Spotify's evolution and academic publications, *e.g.* [6, 9, 13], provided deeper insight into Spotify's technological infrastructure. Moreover, chronologically targeted enquiries with the Internet Archive Wayback Machine helped us to accurately deduce the timing of specific incidents by examining the Spotify website at particular points in time.

3.2 Data Analysis

To analyze our data, we drew upon process analysis methods suitable for explaining "*how things evolve over time and why they evolve in this way*" [12:692] through the sequence of events that lead to an outcome. This entailed an iterative process of moving between intrinsically complex and ambiguous raw process data, an emerging empirical understanding of the developmental process, and the development of theoretical constructs that explained it [12, 17].

First, we sought empirical understanding by moving from raw data to a timeline of empirical incidents meaningful to how the platform had evolved. To that end, we reduced the collected data into a corpus capable of producing the empirical observations relevant to key elements in platform dynamics, *i.e.* architecture, governance, and ecosystem [14, 20, 22]. This entailed removing posts about e.g. competitions, social events at the Spotify office, and music awards, while posts related to interfaces (*e.g.* APIs and SDKs) or apps, changes to rules for use of apps or interfaces, and external actors connected to the platform, *e.g.* license holders, customers, or developers, were seen as relevant for

further analysis. We analyzed the resulting corpus, constituted by 1048 relevant posts, to deduce and chronologically structure incidents [17] (i.e. empirical observations of activities or effects related to the three elements in platform dynamics). For this purpose, we utilized Aeon Timeline software to construct and visualize a timeline of incidents. Each incident was constituted by a title, the date of its occurrence, verbatim descriptions from one or several posts, and a description of how it related to platform architecture, governance, and ecosystem. The timeline and constitutive incidents was triangulated and complemented with data from the other sources mentioned above, resulting in an empirical timeline constituted by 415 discrete incidents. The development of the timeline revealed 3 chronological breakpoints where architectural and governance structures had been distinctly altered, thus constituting different conditions for primarily developers' and end-users' interaction with the platform. As shown in figure 1, the breakpoints also corresponded with increases in subscriber growth.

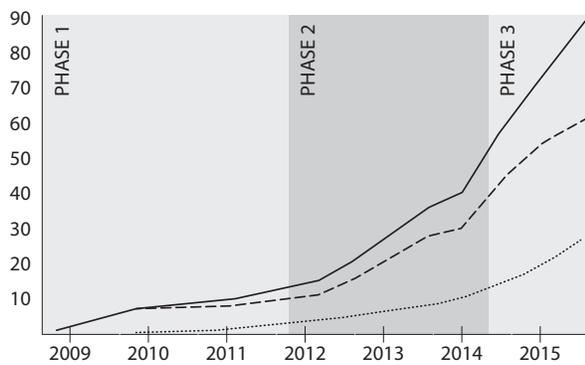


Figure 1. Total, free, and paid subscriber growth in millions over 3 analytical phases.

Second, in order to understand the rationales [20] underlying the identified shifts, we adopted the concept of boundary resources as a vehicle to move from the fine-grained timeline to a generalized, phasic understanding of the process [17]. Understood as software tools and regulations that a platform owner supplies in order to resource and control contributions from external developers and end-users [c.f. 8], the concept was utilized to understand if, how and why each architectural and regulatory change had enabled or constrained the actions of end-users or developers. By treating the identified breakpoints as indicators for phases, we generalized from instances of boundary resource alterations to deduce boundary resource configurations as higher-order constructs temporally delimited by each phase. To distill underlying rationales, we examined what certain configurations

allowed or prohibited developers and end-users to do in relation the actual developmental stage of the platform

4. Boundary resource configurations in the evolution of the Spotify platform

4.1. Prologue

At the beginning of this millennia, the music industry was characterized by plummeting music sales brought on by Internet piracy, but also by a dawning hope for the new medium to generate revenues. On this topic, IFPI reports a 23% decline in global sales between 2000 and 2005, and in 2007, it estimated that for each music track sold, 20 was illegally downloaded. Concurrently, digital sales started to generate the first significant revenues. Following the proliferation of MP3 devices from a wide range of manufacturers, Apple launched its first iPod in 2001. In contrast to its competitors, the iPod was connected to a proprietary service (i.e. iTunes) through which licensed music content could be bought and downloaded. iPods came to control over 50% of the global portable music device market in 2004. With the launch of iPod Touch and the iPhone in 2007, Apple's market position was further strengthened. In contrast, the digital music streaming services that existed at the time were significantly smaller, accounting for a mere 7% of digital music revenues in 2006. As stressed by IFPI, these services suffered from the interoperability that characterized the music industry at the time as specific services and file formats were locked-in to certain devices. As a result, subscription-based streaming services without a proprietary device faced significant problems with attracting users.

While piracy ravaged the music industry and Apple dominated the digital music market, the Spotify company was started in May 2006. In the spring of 2007, the Spotify service became accessible for the first beta-testers in the form of applications for Windows and MacOS. Initially, the Spotify service was offered free of charge and consisted of a basic digital music player with an attached library of links to music files that streamed to the user's computer when clicked. Starting in February 2008, beta-testers began to be subjected to advertisement from third-parties through the service. Soon after, restrictions were applied to ensure that an account only could be used by a single person to enable better targeting for advertisers and the generation of personalized use data.

4.2. Phase 1: Building platform capacities

The first phase was triggered by the official launch of the service in October 2008. At this stage, the primary concern for interface configuration was to improve the platform's capacity to generate and exploit value from its emerging ecosystem. As a first step towards this objective, a "freemium" subscription model was applied, within which new subscribers could choose to register a free or a premium account. Free subscriptions enabled music streaming without monetary payment, but were instead ad-financed, and registering for one required an invite supplied by Spotify. In contrast, anyone could register for a premium account and for a monthly fee use the service without advertising. At the time, Peer-to-Peer (P2P) technology had been implemented within proprietary desktop apps to support central streaming servers and provide better scaling conditions. In effect, P2P activated subscribers, their devices, and internet connections to participate in service distribution, thus mitigating the otherwise directly straining effect on server capacity that each new added subscriber would have. Within a year, higher bitrate sound and offline listening were introduced as additional premium benefits. In October 2011, more restrictions were added to the free subscription that limited listening to 10 hours per month and 5 plays for any individual track after 6 months of unlimited use. At the same time, the invite requirement was removed.

By dividing accounts into segments with different rules associated with them in this way, monetary value appropriation from streaming was initiated while subsidized subscriber growth could be maintained and controlled. At first, control over the amount of invites available implied controlling the upper-limit of free subscribers. By switching to limiting time and plays, the free subscriber-base were allowed to grow while their use of the service could be controlled. As only 9% of streaming traffic for free subscribers came from central servers and the rest was managed by local cache and P2P [11], the growth of free subscribers also implied a relatively limited server strain. Still, converting free subscribers to premium was a key issue, to which end the limitations applied to free accounts and benefits applied to premium served to incentivize subscribers to upgrade their accounts.

During this phase, Spotify extended its service from the two proprietary desktop apps to address other existing and emerging third-party platforms. To this end, proprietary apps for the iOS and Android mobile platforms were made available in September 2009 and were soon followed by apps for Symbian, Windows Phone, and WebOS mobiles, along with apps for several TV and audio devices. As a result, the Spotify service was embedded within at least 15 new devices

during this phase. Whereas the proprietary desktop apps were open to both subscriber tiers, use of the new apps was restricted to premium subscribers only.

The first developer interface, the Libspotify Application Programming Interface (API), was launched in April 2009, and the second, the Metadata API in October. These interfaces did not offer any means for distributing or integrating apps but were instead outbound-directed, i.e. they enabled the diffusion of third-party Spotify apps for external platforms. Libspotify enabled developers to draw upon data, functionality, and to stream content from Spotify. To use the API, external developers required a premium account, apply for an application key, and agree to developer rules that hindered use for commercial purposes. To use third-party apps created with Libspotify, subscribers needed to have a premium account. The Metadata API, on the other hand, did not require premium accounts nor application keys. It was instead limited to providing external developers with data about, and links to, Spotify content. Thus, it did not enable the creation of apps that could stream music. The APIs may thus be considered as outbound developer boundary resources that encouraged the generation of external third-party apps while they simultaneously ensured value appropriation for Spotify. To this end, the digital tools and data provided facilitated the generation of third-party apps, the premium restriction enabled both income and data generation from their use, and data format restrictions implied that third-party apps would in effect direct users to Spotify proprietary apps.

During this phase, capabilities for content curation within proprietary Spotify apps was also gradually improved. In February 2010, the "related artists" feature grouped together lists of similar music based on what tracks subscribers tended to listen to in subsequent order. In April that year, an app update included the ability for subscribers to retrieve data from their Facebook accounts, including demographic and social network data, to extend their Spotify user profiles and to establish social networks within Spotify. With public profiles and the ability to become friends with others, subscribers thus became able to share content and collaboratively edit playlists. Hence, as boundary resources between platform and subscribers, proprietary apps became increasingly able to generate use data that was exploitable by the platform. In feedback loops, new tools for music streaming were provided, their use generated data, and the data was utilized by the platform to provide new tools for content curation. At this point, curation was based on platform-wide use data, and provided generalized recommendations. However, with extended demographic and social data along with new

tools for sharing and organizing content, subscriber boundary resources became able to generate new forms of data that could be utilized further on.

In conclusion, boundary resources were in this phase configured to improve the platform's capacity to generate and exploit value from its surrounding ecosystem. This included fostering a balanced growth of subscribers, enabling mechanisms for monetary value appropriation, and to initiate and utilize tool-data-tool feedback loops. The subsidized use implied by the rules of the free subscription lead to a steady subscriber growth, expanding from a total of 1 million in March 2009, to 10 million in July 2011. Free subscribers were inadequately financed by advertising, yet through P2P technology, the platform could resource them to mitigate their costs, and by applying rules that either limited membership or extent of use, their strain on the platform could be controlled. The proliferation of proprietary apps and the design of developer interfaces as outbound directed show the importance of scaling the service across external devices. This made the service increasingly accessible, but also served to facilitate conversion to the premium tier. Overall, rules associated with subscription tiers served as the key mechanisms to ensure monetary value appropriation from the use of proprietary as well as third-party apps. Finally, analysis of updates to proprietary apps shows how new tools for data generation were continuously embedded within subscriber boundary resources and used to produce new curation tools. As becomes evident in the next phase, the capacities built during this phase were key in attracting more actors to the ecosystem and to incentivize their engagement in accretive behaviors for the platform.

4.3 Phase 2: Exploiting generativity

The second phase started in December 2011 when Spotify announced its internal app store: App Finder. During this phase, platform interfaces were configured to simultaneously attain both generative service diversification and generative scaling across external devices. In achieving the former, the App Finder offered external developers a new API for retrieving data and functionality from the Spotify platform. By combining assets from the API with external assets, developers could now develop apps that could be distributed through Spotify's proprietary apps for Windows or MacOS. However, to be eligible for distribution, apps needed to be approved by Spotify. In contrast to other app stores, the App Finder offered no possibilities for developers to monetize their apps. Despite this unclear incentive structure, the App Finder came to host at least 80 unique applications

from different developers. To subscribers, the App Finder provided the ability to customize the service by installing and using apps that, for example, provided curation in the form of playlists or complementary content related to artists. From a boundary resource perspective, the App Finder may thus be understood as a tool that enabled increased service diversification. In contrast to previous developer interfaces, the App Finder was inbound directed, i.e. it provided means for the integration of apps within the Spotify service.

In terms of scaling the service, at least 13 new proprietary Spotify apps were developed for mobile, audio, and TV devices during this phase, each requiring premium subscriptions to use. In addition, a new interface aimed at external device manufacturers called Spotify Connect (henceforth Connect), was launched in September 2013. Connect enabled developers to embed Spotify compatibility within the firmware of an audio device, thus enabling premium subscribers to use a proprietary Spotify app as a remote control to play Spotify content on the audio device. During this phase, at least 5 manufacturers made their devices Connect compatible. Concurrently, Spotify improved the Libspotify API, focusing primarily on extending its compatibility with external platforms. In March 2014, a new Software Development Kit (SDK) was provided to further facilitate the development of third-party Spotify apps for the iOS platform. This continuous proliferation of apps and integrations within external devices thereby entailed a diffusion of boundary resources that extended access to potential subscribers. With the premium requirement instated for use of any other app than those for desktops, it also served to incite conversion from free subscriptions.

As the number of subscriber interfaces increased, restrictions associated with the free subscription tier were altered. In March 2012, the 5 plays per track limitation was removed, and free subscribers became eligible to limited use of mobile apps in the middle of 2012. Finally, the monthly time limit for free subscriptions was removed in January 2014. Indeed, these actions resulted in a more attractive proposition for free subscribers that may seem contradictory to conversion efforts. However, a large subscriber-base was important to attract key ecosystem actors such as developers and advertisers, to tackle emerging competition in the form of streaming services from Microsoft and Google, and for the generation of valuable use data.

Throughout 2013, proprietary Spotify apps were intensively updated to provide new tools for networking, filtering, curation, and complementary content. First, the "Follow" function enabled subscribers and artists to establish relationships with

each other and automatically retrieve information about each other's activities. The relationships created with Follow was in turn utilized by "Discover" – a tool that provided personalized content recommendations based on a subscriber's network, own past activities within the service, and data from external sources. The external data was provided by three actors that were concurrently providing apps for the App Finder. This approach was replicated when the functionality of an app in the App Finder was implemented in the core service to provide lyrics for tracks. Finally, "Browse" – a tool that provided curation in the form of pre-made playlists based on a diverse set of use data was implemented. Following these implementations, Spotify announced that it would no longer be possible to submit new apps for the App Finder, thus initiating a decommissioning process that ended in the App Finder being removed from the service. From a boundary resource perspective, these actions again demonstrate how adding tools that enabled subscribers to interact with content and actors generated data exploitable in the development of new tools for service improvement. However, they also shine new light on the role of the App Finder. Initially leveraged as a way to diversify the service offering by mobilizing external app developers, it was closed after a limited number of apps had been cherry-picked and integrated within the platform. Without any direct monetary compensation, external developers generated variation by supplying apps on-top-of the core service. Subscribers then selected which apps to install and use, thus providing information about their preferences. In that way, both developers and subscribers could be utilized to generate a decision basis for which functionalities to integrate within the core service. Once integrated, they could be offered as tools within several apps instead of just on the proprietary desktop apps.

Overall, the boundary resource configuration during this phase primarily served to trigger and facilitate a generative expansion of both service scale and scope. By designing new and improving existing outbound developer interfaces, external developers became better able to distribute the Spotify service. While such efforts had previously been directed at the service layer, *i.e.* by providing means for developers to develop apps for external app stores or the web, the Connect interface was aimed directly at external devices. By enabling manufacturers to integrate the service, the number of potential devices to which the Spotify service could be diffused increased significantly. In contrast, the App Finder represented an inbound interface that offered developers the means to both develop and distribute their apps and thus generate service diversity. However, the limited time it was available indicates that the App Finder was

ultimately leveraged as a means to enable informed cherry-picking of tools for core integration based on evidence of subscriber preferences. Further, the design of subscriber boundary resources resulted in increasingly elaborate and personalized feedback mechanisms that drew upon and improved subscriber interfaces. Over the extent of this phase, regulations on free subscriptions were also loosened, thus further enabling subscriber growth, increased service use, and more extensive data generation from use with the potential cost of lowering conversion rates from free to premium. The overall growth of subscribers during this phase was substantial, as Spotify reported to have attained 40 million in total (whereof 10 million premium) in May 2014. Whereas the distribution of subscribers points to a clear dominance of free subscribers, a refined approach for financing their use was implemented during the next phase.

4.4. Phase 3: Consolidating the core service and facilitating generative service scale

The third phase started in March 2014 following the removal of the App Finder. In contrast to the previous phase, the boundary resource configuration now sought to consolidate the service scope across external devices while simultaneously facilitate the generation of external subscriber interfaces. As a first step towards consolidation, in April 2014, Spotify began phasing out the P2P technology that had helped offloading central servers. Only integrated in the desktop apps, P2P was unable to support the growth of streaming to other platforms and streaming to mobile apps were increasing rapidly. Further, the proliferation of 8 new proprietary Spotify apps beyond desktop platforms continued.

In terms of changes to subscriber interfaces, these became homogenized across devices as new features and tools were added. App updates continued to bring new curation features that were based on data generated from subscriber's previous use of tools. For example, a "Top tracks in network" playlist that was implemented in December 2014 generated lists of tracks based on their popularity within the network that a certain subscriber had created with the Follow function. Second, Spotify Running was implemented in May 2015 and offered subscribers a playlist that adapted its content based on their listening history, but also on their running pace. Third, the "Discover weekly" playlist made available in July 2015 provided new personalized content recommendations for content each week. These recommendations were based on an analysis of how all subscribers had been using the playlist tool to organize and group together

tracks. As alterations to boundary resources, these app updates thus replicated the approach of providing tools that subscribers used to organize, share, and create relationships, which in turn generated data that was utilized to provide increasingly personalized matches between content and subscriber.

As capabilities for leveraging user generated data improved, the rules associated with the free subscription tier were changed so as to admit use of more devices, and existing tools for curation were leveraged to better finance free subscribers. After the initial desktop only principle that was extended to include limited use of mobile apps in the previous phase, free subscriptions were now made eligible for using Spotify on PlayStation as well. In light of the increasingly generous value proposition to free subscribers, Spotify announced that external advertisers could target its ad-financed tier not only through demographic data provided by information in user profiles, but also based on what they were doing and how they were feeling. Such detailed targeting was enabled through subscribers using the mood- or activity-based playlists that had been previously implemented as elements of the Browse tool. Hence, from a boundary resource perspective, this shows how subscriber's use of tools for content curation provided the platform with data that could be used to improve the value proposition offered to another ecosystem actor.

Subscriber interfaces aside, the focus of developer interfaces during this phase was exclusively outbound directed. First, at least 5 new devices were made Connect compatible during this phase. Second, a new Web API was launched in June 2014 to better support the development of third-party Spotify apps for the web. Replacing the old Metadata API, the Web API initially enabled external apps to retrieve extensive metadata, album art, user profile information, as well as short previews of audio content. Third, a new Android SDK was made available shortly after, which together with the already available iOS SDK came to constitute a more focused offering aimed at the two external mobile platforms. Subsequent to incremental improvements to extend the diversity of data retrievable through the new SDKs, the legacy Libspotify API was terminated.

The new API and SDKs came with new rules. As with the previous Metadata API, the Web API was open for all to use, yet the form in which content could be retrieved was limited. Although it allowed streaming of audio content to third-party apps, streamed tracks were limited to 30 second previews. This limitation was again utilized in December 2014 as Spotify announced that apps created with the new SDKs could be monetized, as long as they did not offer

streaming of full-length tracks. If third-party apps that could stream full-length tracks were developed, these could not be monetized by third-parties, but would instead only be usable by premium subscribers. These developer interfaces could thus be seen as outbound boundary resources primarily designed to generate subscriber interfaces across external devices. In particular, collapsing the generic Libspotify into two SDKs indicate a more focused steering of developers towards devices running iOS and Android. Although limited, the possibilities for third-party monetization provided further incentive for external developers to create and distribute new subscriber interfaces. With premium requirements on use, or through limitations on content retrievable through interfaces, Spotify ensured that it would appropriate value as a platform owner.

At the end of this phase, Spotify counted close to 90 million subscribers in total, out of which around 30 million were premium accounts. Reflecting these numbers, in December 2015, app market analysis firm App Annie announced that Spotify had become the number one digital music streaming service in the world in terms of active users, app downloads and revenue. As a final stage towards reaching this position, the boundary resources in this phase can be understood as configured for service consolidation and generative external service distribution. Thus, subscriber interfaces were gradually improved with new tools for organizing, networking and sharing, which while providing a value in their own, also generated data that was used to calibrate increasingly personalized curation tools. As indicated by the removal of the App Finder, the P2P system, and many of the restrictions for free subscribers, service consolidation also implied developing a more unitary service offering across subscriber segments and devices. In contrast to the previous phase, developer boundary resources were unidirectional, designed for the outbound diffusion of data and functionalities without any means for distribution within the core service. As such, they did not primarily intend to extend the service scope, but to facilitate a generative extension of service scale by encouraging external developers to distribute apps across external devices. By employing limitations on content or applying demands for premium subscriptions, Spotify ensured that external apps created with its boundary resources would either attract users to proprietary apps, or generate direct monetary value.

5. Discussion

Extant research on digital platform evolution has predominantly investigated the distinct subset of digital *product* platforms as constituted by bundles of a proprietary device, OS, and associated app store [e.g. 2, 3, 4, 5, 8]. Albeit having generated significant contributions in detailing how and why such platforms evolve and succeed, current research is limited in addressing digital *service* platforms, *i.e.* platforms that are exclusively positioned within the service layer of industry architectures and hence are void of proprietary devices. To address this shortcoming, we investigated the evolution of the Spotify platform as a prototypical example of a digital service platform in order to detail architectural characteristics and their strategic influence across its evolutionary process. Against this backdrop, our study has three contributions.

First, our analysis shows that the architectural nature of digital service platforms produces unique rationales for leveraging interfaces to external developers, which suggests a reconsideration of the unidirectional nature of boundary resources in digital platform evolution. Boundary resources originate in the idea that platform owners can supply and regulate interfaces that enable external developers to create and distribute an inbound flow of modules on top of a platform, thereby extending its functional scope and diversity [8]. While our analysis revealed that this fundamental logic of inbound scope extension was present in the evolutionary process of the Spotify platform, it only occurred during a limited timeframe (phase 2). Not having a proprietary device, Spotify instead needed to align with the boundary resources of multiple external digital product platforms to enable distribution of its service. Therefore, maintaining a service diversity that was strictly controlled rather than dynamically generative was more viable for ensuring technological compatibility and compliance to externally set rules. More importantly, the lack of a proprietary device from which Spotify could draw hardware and OS capabilities implied that the upper-limit for diversity that could be provided by external developers was reached rather quickly. These architectural conditions jointly suggest that while digital service platforms may indeed leverage inbound boundary resources to attain variety on top of a core service, limitations in technical and logical capabilities along with dependencies to external platforms will likely render such efforts unfruitful in the long run. In contrast, the essential developer boundary resources for the Spotify platform were instead outbound directed. They were comprised of tools that enabled the development of Spotify apps and integrations for external devices (*i.e.* not extending the core service functionality per se but rather extending access to it)

along with rules to ensure value could be appropriated by the platform-owner. For example, the introduction of Spotify Connect and the launch of several new APIs during phase 3 primarily served the purpose of stimulating a controlled outbound generativity aimed at diffusing the service on multiple devices and hardware while at the same time funneling users to Spotify's proprietary apps by clear regulatory frames for what external apps were allowed to do. Our findings thus suggest that while digital product platforms principally rely on inbound developer boundary resources to attain a generative yet controlled extension of diversity and scope, the key developer boundary resources for digital service platforms are outbound directed, aimed at mobilizing external developers for generative scaling.

Second, our study illuminates the important and under-theorized role of end-users in digital platform evolution, and by extension, how the boundary resource concept fruitfully may be utilized for this purpose. We explored end-user boundary resources by tracing functional changes in proprietary Spotify apps along with their associated rules. For example, as proprietary apps were continuously updated with tools that enabled end-users to organize, share and filter content along with establishing social networks, core data was unremittingly generated and utilized by Spotify to progressively improve the platforms precision in terms of value offering. Other examples of end-user resourcing included the embedment of P2P technology, which mobilized end-users to actively participate in streaming distribution, and the App Finder to engage them in screening functionalities to determine future core integrations. Moreover, regulations applied through the subscription model served the purpose of generating personalized use-data, enabling control over end-user growth, guiding end-users to specific devices, and ultimately fueling conversion from free to paid subscriptions. Overall, our analysis demonstrates how Spotify spent considerable effort in fine-tuning end-user boundary resources as a means to improve and adjust its overall evolutionary trajectory.

As a final and aggregated contribution, our study reveals a number of key architectural characteristics to digital service platforms, suggesting distinct evolutionary patterns from those of digital product platforms. First, the absence of a proprietary device implies that digital service platforms have limited internal assets from the outset that can be offered through interfaces and as a result, assets need to be built over time through cumulatively enabling and analyzing transactions. Second, digital service platforms have no *a priori* specified proprietary distribution channel for reaching end-users, meaning

that they depend on external device providers for service distribution. Third, being void of a proprietary device implies no obligations to create or maintain positive feedback mechanism between it and the service, which suggests the possibility to instead exploit device multi-homing to create lock-in effects through ubiquity.

Although one focal platform remained the focus of our study, our findings also shine light on the core dependencies that may arise between digital platforms. In common with other recent contributions [5, 18], our study therefore emphasizes inter-platform dynamics as an interesting avenue for future research. As platforms become platforms for platforms, new and interesting research questions related to platform architectures (*e.g.* how may digital service platform architectures be designed to enable scaling across multiple external platforms?), governance (*e.g.* who controls data and monetary value appropriation and what are the consequences?), and ecosystems (*e.g.* how can a service platform-owner avoid or manage potentially predatory actions from a product platform provider?). Moreover, our findings indicate that more research is needed to understand a wider variety of ecosystem actors may add to the dynamics of digital platforms, as compared to the current dominant focus on platform owners and third-party developers.

6. References

- [1] Barrett, M., E. Davidson, J. Prabhu, and S.L. Vargo, "Service Innovation in the Digital Age: Key Contributions and Future Directions", *MIS Quarterly* 39(1), 2015, pp. 135–154.
- [2] Basole, R.C., "Visualization of interfirm relations in a converging mobile ecosystem", *Journal of Information Technology* 24(2), 2009, pp. 144–159.
- [3] Basole, R.C., and J. Karla, "On the Evolution of Mobile Platform Ecosystem Structure and Strategy", *Business and Information Systems Engineering* 3(5), 2011, pp. 313–322.
- [4] Boudreau, K.J., "Let a Thousand Flowers Bloom? An Early Look at Large Numbers of Software App Developers and Patterns of Innovation", *Organization Science* 23(5), 2012, pp. 1409–1427.
- [5] Eaton, B., S. Elauf-Calderwood, C. Sørensen, and Y. Yoo, "Distributed Tuning of Boundary Resources: The Case of Apple's iOS Service System", *MIS Quarterly* 39(1), 2015, pp. 217–243.
- [6] Ellis, M., S.D. Strowes, and C. Perkins, "An Experimental Study of Client-Side Spotify Peering Behaviour", *IEEE Conference on Local Computer Networks*, (2011), 267–270.
- [7] Gawer, A., "Bridging differing perspectives on technological platforms: Toward an integrative framework", *Research Policy* 43(7), 2014, pp. 1239–1249.
- [8] Ghazawneh, A., and O. Henfridsson, "Balancing platform control and external contribution in third-party development: the boundary resources model", *Information Systems Journal* 23(2), 2013, pp. 173–192.
- [9] Goldmann, M., and G. Kreitz, "Measurements on the spotify peer-assisted music-on-demand streaming system", *IEEE International Conference on P2P Computing*, (2011), 206–211.
- [10] Huang, J., O. Henfridsson, M.J. Liu, and S. Newell, "Growing on Steroids: Rapidly Scaling the User Base of Digital Ventures Through Digital Innovation", *MIS Quarterly* 41(1), 2017, pp. 301–314.
- [11] Kreitz, G., and F. Niemela, "Spotify—large scale, low latency, P2P music-on-demand streaming", *IEEE International Conference on P2P Computing*, (2010), 1–10.
- [12] Langley, A., "Strategies for Theorizing from Process Data", *The Academy of Management Review* 24(4), 1999, pp. 691–710.
- [13] Lee, J., and N. Berente, "Digital Innovation and the Division of Innovative Labor: Digital Controls in the Automotive Industry", *Organization Science* 23(5), 2012, pp. 1428–1447.
- [14] Lusch, R.F., and S. Nambisan, "Service Innovation: A Service-Dominant Logic Perspective", *MIS Quarterly* 39(1), 2015, pp. 155–175.
- [15] Ostrom, A.L., M.J. Bitner, S.W. Brown, et al., "Moving Forward and Making a Difference: Research Priorities for the Science of Service", *Journal of Service Research* 13(1), 2010, pp. 4–36.
- [16] Ostrom, A.L., A. Parasuraman, D.E. Bowen, L. Patricio, and C.A. Voss, "Service Research Priorities in a Rapidly Changing Context", *Journal of Service Research* 18(2), 2015, pp. 127–159.
- [17] Poole, M.S., A.H.V. de Ven, K. Dooley, and M.E. Holmes, *Organizational Change and Innovation Processes: Theory and Methods for Research*, Oxford University Press, 2000.
- [18] de Reuver, M., C. Sørensen, and R.C. Basole, "The digital platform: a research agenda", *Journal of Information Technology*, 2017.
- [19] Scherer, A., N.V. Wunderlich, and F. von Wangenheim, "The Value of Self-Service: Long-Term Effects of Technology-Based Self-Service Usage on Customer Retention", *MIS Quarterly* 39(1), 2015, pp. 177–200.
- [20] Thomas, L.D.W., E. Autio, and D.M. Gann, "Architectural Leverage: Putting Platforms in Context", *Academy of Management Perspectives* 28(2), 2014, pp. 198–219.
- [21] Tilson, D., K. Lyytinen, and C. Sorensen, "Research Commentary--Digital Infrastructures: The Missing IS Research Agenda", *Information Systems Research* 21(4), 2010, pp. 748–759.
- [22] Tiwana, A., B. Konsynski, and A.A. Bush, "Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics", *Information Systems Research* 21(4), 2010, pp. 675–687.
- [23] Yin, R.K., *Case Study Research: Design and Methods*, SAGE Publications, 2013.
- [24] Yoo, Y., O. Henfridsson, and K. Lyytinen, "The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research", *Information Systems Research* 21(4), 2010, pp. 724–735.