Article

SM+S social media + society

Social Media + Society
October-December 2020: 1–13

DOI: 10.1177/2056305120971632
journals.sagepub.com/home/sms
SAGE

The Material Conditions of Platforms: Monopolization Through Decentralization

Tobias Blanke[1] and Jennifer Pybus[2]

Abstract

In this article, we add research on technical integration and dependency to the theories of platformization. Our research seeks to understand how platforms have been able to technically integrate themselves into the fabric of the mobile ecosystem, transforming the economic dynamics that allow these largely enclosed entities to compete. We therefore want to consider platforms as service assemblages to account for the material ways in which they have decomposed and recomposed themselves for developers, enabling them to shift the economic dynamics of competition and monopolization in their favor. This article will argue that this shift in the formation of platform monopolies is being brought about by the decentralization of these services, leading to an overall technical integration of the largest digital platform such as Facebook and Google into the source code of almost all apps. We present new digital methodologies to surface these relations and material conditions of platforms. These methodologies offer us a whole new toolkit to investigate how decentralized services depend on each other and how new power relations are formed.

Keywords

platformization, decentralization, digital methods, mobile ecosystem

Introduction

Platforms can no longer be understood as single monolithic applications. For Aradau et al. (2019), the digital materiality of platforms is foregrounded in the process wherein they are broken apart into services and reassembled into new products. This transformation has been slowly taking shape for the past few years. Amazon, e.g., famously discovered that it could extend itself beyond selling books by selling computing services in the Cloud, which accounted for almost 13% of their revenue in 2019 (Protalinski, 2019). This new kind of extension or platformization (Blanke, 2014; Helmond, 2015) has allowed these companies to become a dominant and constitutive part of the web's infrastructure and economic landscape. Their governance is thus concerned with the control, stabilization, and extension of the means that allow these entities to expand via their capacity to de/recompose their existing infrastructures. We can see this with AirBnB's software development model, in which their stack was divided into a number of distributed services rather than one singular application (Datadog, 2018). At the end of this infrastructural decomposition, AirBnB recomposed itself into a multitude of interconnected services, a process that has come to define platforms.

As we shall argue, the digital materiality of platforms is defined by this reduction into reassembled elements. A new field called "platform studies" (Plantin et al., 2018) has recently focused on this relationality and subsequent tensions that exist between the open infrastructures of the web and the enclosed walled environments of platforms. In looking at their political economy, the enclosed walls are first and foremost monopolies trying to dominate the open and mobile web, a tendency that Srnicek (2017) warns is "built into [their] DNA." Subsequently, Van Dijck et al. (2018) do not see these large-scale entities as single applications but rather argue that they should be understood as ecosystems that bring users together with a number of private and public actors. Bratton (2015) takes a similar political–economic perspective, but instead looks to the expansive geopolitical role that platforms play as both computational apparatuses and governing architectures. Ultimately, the concern with platform monopolies has led to demands for greater transparency and accountability

[1]University of Amsterdam, The Netherlands
[2]King's College London, UK

Corresponding Author:
Tobias Blanke, Institute of Logic, Language and Computation, University of Amsterdam, Spui 21,1012 WX Amsterdam, Netherlands.
Email: t.blanke@uva.nl

over these ever-expanding infrastructures. Given the opacity that is built into the myriad ways in which data flow through platforms, Gillespie (2018) warns against their monopolization by focusing on the struggles over how platforms organize our content, as do Rieder and Sire (2014), who examine how large platforms like Google have an "incentive to bias" toward themselves.

While we do not disagree with this platform logic or propensity toward monopolization, our research seeks to understand how platforms have been able to technically integrate themselves into the fabric of the mobile ecosystem, transforming the economic dynamics that allow these largely enclosed entities to compete. Subsequently, we want to consider platforms as service assemblages to account for the material ways in which they have decomposed and recomposed themselves, enabling them to shift the economic dynamics of competition and monopolization in their favor.

For us, platform monopolization is extended by questioning how we are to understand the services that companies like Google and Facebook provide for the development of apps. These include anything from digital analytics, to advertising, to social sharing, to data trends, to insights, to crash reporting, and so on. This technical integration is open to a number of actors who are invited to embed an array of distributed so-called value-added services of de- and recomposed platforms. Ultimately, this has led to a much deeper technical integration of these ecosystems, which has arguably been overlooked and underresearched. We therefore seek to put forward new methodologies to make visible the "intermediation" between services and platform infrastructures (Anderson & Blanke, 2015).

To examine platforms and develop new methodologies, we have had several research projects since 2014, which have been the basis for the empirical work in this article. These have concentrated on the inner workings of the global digital services ecosystem constituted by mobile phone apps. In 2014, we started with the "Our Data Ourselves" project, which investigated how young people are tracked by apps when they use their mobile phones (Blanke et al., 2014). We developed a toolkit to trace ingoing and outgoing communications of all apps installed on Android mobile phones. This project included investigations into how young coders could (re)shape their own data through a series of hackathons (Pybus et al., 2015) and then later to consider how participatory workshops can be used as an exploratory research method, which Coté and Pybus (2016) have called the "techno-cultural method." The latest project in this series, "Zones of Data Translation" worked with the data NGO Tactical Tech to embed our tools and methodologies into their exhibitions and public events.

Integrating Infrastructures: The Rise of Value-Added Services

Our forensic inspection of mobile applications reveals that they comprise mainly decentralized components or services, often coming from platforms such as Google and Facebook but also from a vast number of smaller contributors. We have covered several areas where we observed similar patterns. Our work with refugee apps (Aradau et al., 2019), for example, demonstrated that digital humanitarianism apps rely on a number of value-added services. For the user, this means that even when then they are logged out of the Facebooks and Googles, they are probably still logged into their service ecosystem via at least one of its developer tools, which have been integrated in over 45% of mobile applications (Binns et al., 2018).

The success of Google and Facebook's integration into the wider mobile ecosystem begins with the fact that developers often rely on third-party services as the most cost-effective way to monetize their apps. This logic is explored by Braun (2013), who examines the business-to-business integration of third parties, focusing on the role of what he calls "transparent intermediaries" or rather those invisible actors that extend and maintain infrastructures of the ad tech ecosystem. Nieborg and Poell (2018) add to this discussion with their concept of "platform dependence," drawing attention to the symbiotic relationship between the content of producers and platforms and, more importantly, to the ways in which these digital infrastructures have made themselves integral to industries that once existed outside their grasp. For this reason, as we shall see, large digital advertising platforms have diversified their in-house development expertise and begun offering services at scale across the mobile ecosystem. In doing so, they are "helping" developers maximize the surplus value they can extract from their apps, while simultaneously expanding their own service infrastructure. These symbiotic, large-scale relationships rely on the technological integration of myriad activities, a point also made by Helmond et al. (2019) in their detailed analysis of Facebook's historical development.

Platform capitalism has unleashed a new dynamic of ever-increasing competition and monopolies as well as technological integration where industries now depend on each other, despite their competition for new customers, lower material, and costs. As we start to closely examine this phenomenon, we can observe that one of the dominant modes of production is dependent on different modalities of monetization. Thus, we seek ways of researching this technical integration and dependency of platform as service assemblages to account for the different constellations of actors and the infrastructure that supports them within the mobile ecosystem.

We see a growing need for a new methodological approach to examine the distributed means by which digital platforms such as Google and Facebook are using the creation of services to further integrate and instantiate themselves within mobile apps to reveal their complex inter-dependency and dependency on these technologies. For app developers, these often "free" services are packaged in what are commonly referred to as software development kits (SDKs). Given the frequency in which these third-party SDKs consistently appear in the majority of mobile applications, we see a symbiotic but also uneven relationship between the services being offered through SDKs and the monetization of apps. The challenge, however, remains in how to access and study this relationship.

Platform studies has started this important work, with many scholars turning their attention to those industries that capitalize on the (re)production and capture of personal data. Srnicek (2017) puts these processes at the center of the business model of platform capitalism. Helmond addresses these questions by considering how data on the web become "platform ready" (Helmond, 2015). Similarly, Zuboff (2019) argues through her conceptualization of "behavourial surplus" that "[Google] discovered a way to profit from its transactions with its real customers: advertisers" (p.182). In short, there is a drive to reveal a user's entire social life online, creating what marketers describe as the "customer journey," that is, the unique pathway that we take through our devices, apps, and clicks to determine those opportune moments wherein we can be reached (McStay, 2018). Here, scholarship on platforms has largely been concentrated on the monetization of user data as the condition of growth within the digital ecosystem. However, what is commonly overlooked is the underlying and integrated network of technical actors that supports the capture of user data well outside the discrete boundaries of the platform.

To capture the long tail of user interactions, new digital industries have proliferated through the creation of services, which provided the conditions of growth for the whole mobile ecosystem. Facebook and Google are unsurprisingly the champions of the new app economy, but others are emerging, too, as we will see Importantly, these platforms do not offer one service through their SDKs, but integrate several, all of which provide unique capacities. For example, Facebook has Facebook Login, Facebook Share, Facebook Analytics, Facebook Ads, and Facebook Places. Similarly, Google offers Google Firebase Analytics, Google Ads, Google DoubleClick, Google Crashlytics, and Google Analytics. These are not anonymous third parties, passively gathering data from our apps, but known services provided by platforms that create new conditions of economic growth and dependency, which expand every time one of these gets integrated into an app. As a result, these platforms maximize data flows and service interactions at a microscopic, infrastructural level of technical integration which makes the entire mobile digital ecosystem complicit in this endless value-added exchange.

At the same time, the Googles and Facebooks have a unique position of power in the ecosystem, as without them it could not grow as a whole. As mature industries, digital platforms reproduce themselves by creating an array of digital services that extend their capacity to profit. However, this infrastructure is not evenly distributed. Some digital industries are more prominent than others, as we will discuss. How can we investigate such a system of dependencies? What kind of measures and metrics can we put in place? The research on ecosystems, for instance, seems to rely often on networks as a model to present inter-dependencies (Blanke, 2014). Networks are indeed useful to present relationships. In this article, we take a different approach and work with co-occurrence and transitivity to model the technical integration in mobile ecosystems. We ask which entities appear together and how they are related to further entities connected to them by separate relationships.

Co-occurrence has been used for the analysis of social relations in Baldassarri (2009), to describe collective actions and model "actors (participants) and topics (content) in online debates" (Hellsten & Leydesdorff, 2020) or as an indicator for a general "relational sociology" (Fuhse, 2015). Co-occurrence helps public healthcare research to derive from social media data sentiments and "consumer health" indicators (Jiang & Yang, 2013). The cited authors agree that co-occurrence in and of itself is not enough to establish dependencies and further analysis is needed. In this article, we thus add more comparison techniques such as outliers and random walks. Overall, co-occurrence is a popular instrument to analyze large amounts of data because it is fairly simple to use and a good indicator of whether two objects of interest belong together. For example, Burnap et al. (2014) employ "co-occurrence of a URL and a hashtag in a tweet" to analyze social media reactions to historical events.

Co-occurrence has been widely used in linguistics for a long time to describe the dependencies of terms and mutual information in texts (Harris, 1957). A simple example is bigrams, where two terms co-occur directly next to each other and indicate a possible semantic relationship if they appear often enough together. Through co-occurrence, linguistics could create more semantic relationships with fairly simple means. When it comes to apps, we are interested in how this might apply to different services that co-occur in their respective code. Just as co-occurring terms describe proximities and dependencies across texts, we will use co-occurring services as a measure of proximity of intentions and dominance within the app ecosystem.

Co-occurrence will provide us with a powerful toolkit to understanding the level of technical integration across the mobile ecosystem. To us, the most powerful digital industries are those that are providing the most commonly embedded services and tools to facilitate the monetization of the developer's app. Subsequently, while the occurrence of discrete third-party services in apps is important, it is the co-occurrence of these services that provide a deeper semantic insight into the reproduction of the mobile ecosystem's infrastructure. With random walks and association analysis in the section "Multiplicities of Services," we finally go beyond how two services co-occur and discover service multiplicities as well as which new services are emerging out of the shadows of the already dominating ones.

Background: Apps and Their Services

The project "Zones of Data Translation (ZDT)" was the latest in a number of projects we had in recent years to analyze the mobile ecosystem. While our previous projects were more focused on particular groups or subjects, ZDT took a broader approach and investigated almost 7,000 Android manifest files for the permissions and services links they include. An Android manifest contains the structure and metadata of an app, its components, as well as requirements. Our work created a sizable dataset consisting of over 45,000 records of services that co-occur in apps according to the manifest data. We analyzed over 5,600 apps, which were taken from a curated set of three sources. We prioritized apps featured on Google Play's most downloaded app list but also included apps from the US Haystack Project (Dance et al., 2018) and the European Exodus Privacy Project (Exodus Privacy, 2020). Together our apps are using 194 SDKs from different third parties between them. On average, a particular SDK appears in less than 5% of apps. Their distribution is highly uneven, as 95% of all SDKs do not appear in more than 10% of all apps. Their connectivity generally does not follow a normal distribution but a power law one, where only a few super-SDKs are used for most of the connections.

The linkage between apps and SDK services shows how important these are for enabling processes of monetization. For example, on average, the apps we analyzed have at least 2.5 integrated third-party SDKs: 20% of apps include at least four SDKs and 10% use six and more. At the upper extreme, there are some further outliers, including "Ready Set Holiday" and "Yeni Milyoner," which include over 32 SDKs. Figure 1 visualizes the SDK density per app.

Before we move to an analysis of co-occurrence let us continue with a few further basic statistics of these SDKs. The most commonly appearing SDKs stem from Google and Facebook, which dominate the ecosystem. Google Firebase Analytics, for instance, can be found in almost 70% of all apps according to Figure 2.

Apps and the integrated services they rely on form together a bipartite network. Table 1 is an extract of the network's matrix that lists the number of times a third-party SDK appears in an app. The first three combinations of apps (rows) and SDKs (columns) from the network's adjacency matrix are listed.

The big internet companies and, in particular, Facebook and Google dominate the network of SDK services. For example, the 10 highest ranked nodes in the bipartite network in terms of betweenness come from these two platforms, as do the 9 highest ranked in terms of closeness. Betweenness measures how many times a node is on the shortest path between other nodes in the bipartite network. Closeness describes how many steps are required to access every other node from a given node. In the two betweenness graphs of the

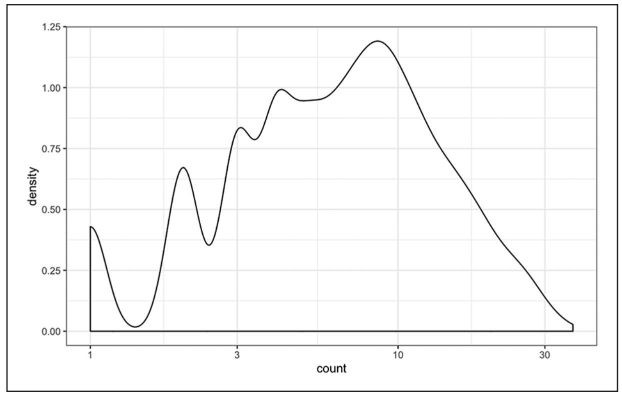

Figure 1. SDK density per app.

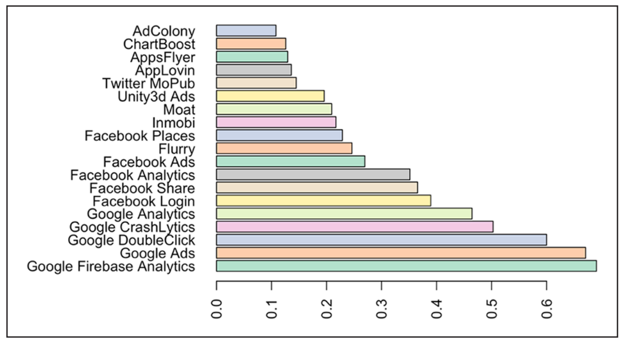

Figure 2. Most frequent SDKs.

Table 1. SDKs per App.

	Google Firebase Analytics	New Relic	Demdex
agl.digital.mobile	1	1	1
air.au.com.minimega.bonza	1	0	0
air.bftv.larryABCs	1	0	0

bipartite SDK-app network in Figure 3, Facebook and Google are the red dots at the center; all other dots tend to be more at the periphery, including the other American internet giants (Amazon, Apple, Microsoft—yellow dots) or Chinese giants (Baidu, Alibaba, Tencent—lightblue).

To continue with the investigation into the importance of Google and Facebook, we introduce four axes of comparisons that concentrate on co-occurrences. First, we present specific data for Google and Facebook. Second, we compare the prevalence of Google and Facebook's SDKs with those belonging to other big US internet companies: Amazon, Apple, Twitter, and Microsoft. Third, we look at their Chinese competitors: Baidu, Alibaba, and Tencent. The fourth category is formed by all the other SDKs, which constitute an array of other services that have been built to support monetization or offer alternatives to the larger platforms.

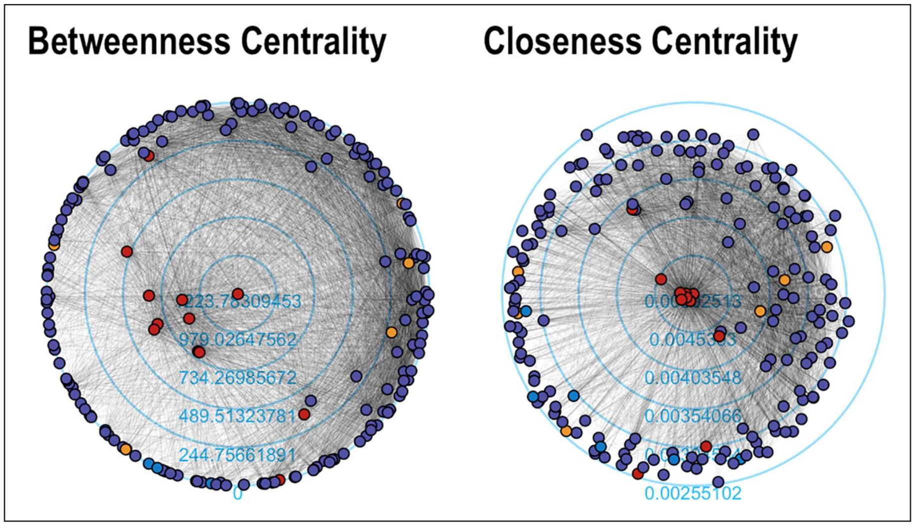

Figure 3. Centrality graphs.

Table 2. Co-occurrences of SDKs.

	Google Firebase Analytics	New Relic	Demdex
Google Firebase Analytics	0	126	199
New Relic	126	0	37
Demdex	199	37	0

Table 3. Co-occurrences of SDKs by Frequency.

Service 1	Service 2	Frequency
Google Ads	Google DoubleClick	3,368
Google Ads	Google Firebase Analytics	2,503
Google CrashLytics	Google Firebase Analytics	2,444
Google DoubleClick	Google Firebase Analytics	2,240
Facebook Share	Facebook Login	2,053

Investigation of Technical Integration Through Co-occurrence

To move to a more "semantically rich" description of technical integration, it is not enough to look at SDKs in isolation. We need to consider how they co-occur and form communities to demonstrate how internet companies are technically dependent. To this end, we take the dot-product of our bipartite matrix with its own transpose and arrive at a count of how services co-occur in Table 2.

Through this count of the co-occurrence of two SDK services, it becomes clear that the big internet companies not only refer to themselves but also directly to each other and are as technically dependent on each other as others are on them. The 20 most common co-occurrences of SDKs in the same app are all reserved for Google and Facebook. Table 3 shows the top five and their frequency.

If we rank how two SDK collections appear together according to their frequency, we find the first combinations wherein neither Google nor Facebook are included at ranks 47 and 64. "InMobi," an Indian global mobile advertising technology company providing contextually relevant ads, co-occurs 950 times at rank 47 with "Flurry," an American mobile analytics, monetization and advertising company owned by Yahoo! "InMobi" and "Moat," a monetization company owned by Oracle focused on brand security, co-occur 704 times at rank 64. Figure 4 shows these ranks with the dotted lines at rank 47 and 64. While both companies are directly competing with Google and Facebook, Table 3 demonstrates that their families of SDKs remain the most frequently integrated.

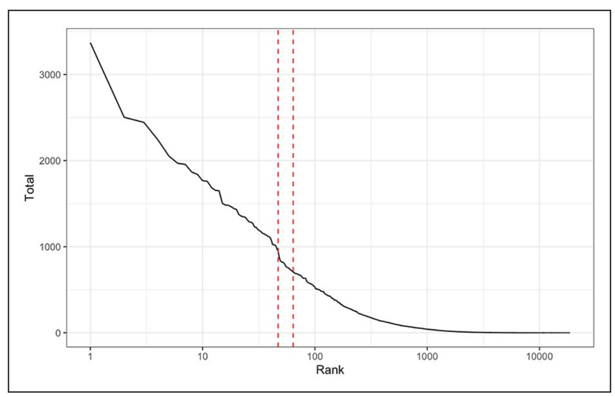

Figure 4. Co-occurrence ranks.

Looking at the co-occurrence of SDKs demonstrates how technically integrated internet companies are. Their SDKs have to work together. If one of them fails, an app will stop working or be at least less powerful with a likely negative impact on all services involved. For instance, Facebook Login

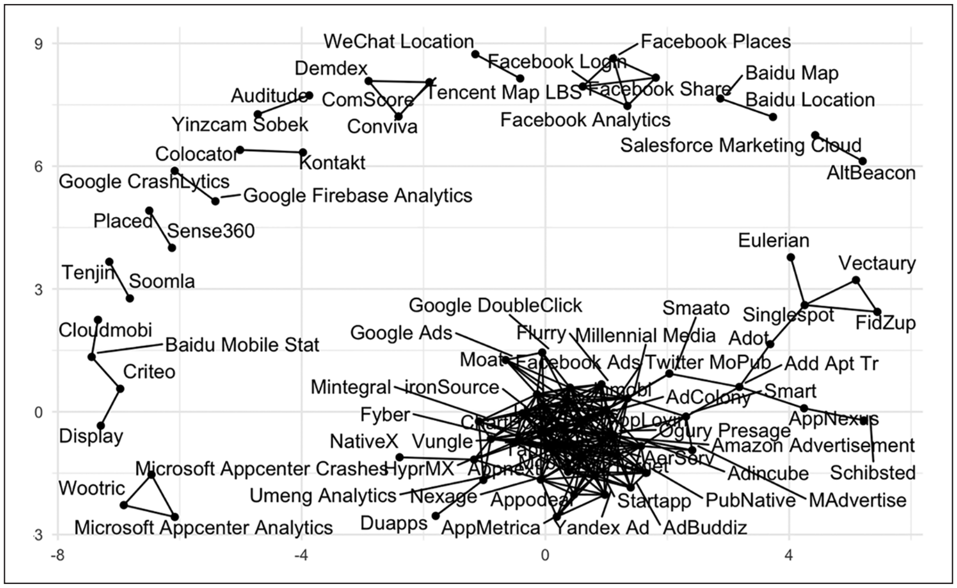

Figure 5. Correlations links between SDKs.

and Google Ad co-occur frequently in apps. If Google Ad would disappear, the app would likely lose a significant source of income and Facebook Login would be able to collect less user information. This describes an involuntary technical integration of SDKs compared with the planned alliances that are generally considered when we are talking about digital ecosystems. Take the well-known example of the alliance between Netflix and Facebook. Shortly after it launched as a digital platform, Netflix added the possibility to log in to access its content via Facebook's identity provider services. This meant it could rely on the large social graph of Facebook users to recommend Netflix content to each other, while Facebook users had content to share and talk about, which kept them further on the site. In 2018, the *NYTimes* reported how far Facebook went to allow select few content providers such as Netflix and Spotify intimate access to their users' private messages (Dance et al., 2018). The principle of a digital ecosystem is thus this privileged mutual agreement to benefit all competing platforms. This same logic applies to SDKs; however, these are invisible and sometimes involuntary alliances, which signal the advanced technical dependencies that exist within mobile applications.

To consider which third parties dominate and create dependencies, we will next expand the semantics of co-occurrence of services in apps from counts to correlations. Compared with the previous simple count of SDKs, their correlation in apps will help us to get a numerical representation of the strength of their technical integration. Then we can, for example, create networks of co-occurrence with edges that depend on the strength of the co-occurrence. The network in Figure 5 shows the links between SDKs that are at least correlated by a factor of 0.3. Service groups of the same kind seem to appear together. According to the pairwise correlation analysis, Facebook services, Microsoft services, Baidu services, and so on all prefer other services from their own background.

At the center of this correlation network is a bulk of SDKs that are technically integrated in what we can see in Figure 5 as a "hairball" of connections in the bottom-right corner. Otherwise, services of the same kind come together. In the bottom-left corner of Figure 5, Wootrich is connected to various Microsoft Appcenter services. Wootrich is a customer service company, which according to its correlations makes heavy use of Microsoft services to deploy its apps. Tenjin, however, is connected to Soomla in the center-left of Figure 5. Tenjin provides services for user acquisition and monetization in the game industry. Soomla offers in-app advertising services that are widely used in the gaming industry. At the top part of Figure 5, we find at the center an example from China. WeChat Location and Tencent Map are connected demonstrating a typical combination of services by in-house development given that WeChat is a Tencent service.

Network visualizations such as those in Figure 5 are useful to map alliances but do not quantify the communities of co-occurrence very well. A small change in the correlation factor can fundamentally change, for instance, the network correlation visualization. We turn therefore to a cluster analysis with k-means to find SDKs that appear together (Blanke & Aradau, 2019). Clustering is a good method to find out more about hidden communities of SDKs, as we try to move away from the predefined big obvious associations with the internet giants. K-means is very useful to discover such hidden (technical) integrations. Its disadvantage is that we need to first determine the correct number of clusters k. According to the Calinski–Harabasz index, our optimal number of clusters is 5, visualized in Figure 6 as five different colors of

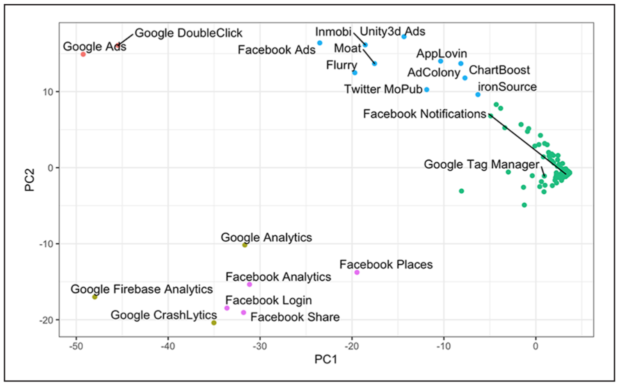

Figure 6. K-means clusters of SDK co-occurrence.

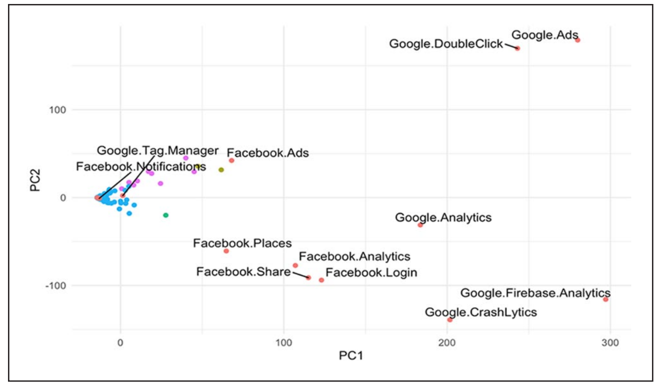

Figure 7. Facebook and Google as outliers.

nodes. The graph is based on a mapping of the five clusters onto a two-dimensional space using a principal component analysis (PCA). This mapping allows us to visualize in two dimensions the separation of SDK clusters as corners in Figure 6.

Considering the overall distribution of clusters across the two PCA axes, we can see that some of the Facebook and Google SDKs are closer to the rest of the services but most of them appear in separate clusters. The cluster in the top-left corner is clearly the Google financial support services for apps, while the Google cluster in the bottom-left corner are Google (analytics) tools for general-purpose use. These are very close to their Facebook equivalents, which appear also in the bottom-left corner. Together, they are also the clusters that are most clearly differentiated. In the top-right corner, we find the Facebook Ads embedded in a family of similar services such as AdColony, Moat, or Flurry. All of these are major providers of marketing services. On the contrary, Google Tag Manager and Facebook Notifications seem to prefer the company of the rest of the apps, which they co-occur most commonly with. They are deeply integrated with the rest of the apps. Google Tag Manager helps manage and deploy marketing tags. Facebook Notifications are updates about activity on Facebook. Both of them are free to use and are close to the main interest of the app ecosystem to provide marketing revenue and thus allow for higher margins. This is why they are so tightly integrated with the other SDKs and stand out from the rest of the Facebook and Google ones.

Our aim has been first to identify those digital industries without whose technical services the others cannot do, and second, to find out how these technical services integrate. We first looked at all the internet giants and could see that they definitely stand out from the rest as they dominate the co-occurrence of services. Among them, Google and Facebook clearly dominate what can be built in the app ecosystem. These companies are special as they allow all others to grow their businesses. To investigate in more detail, we removed Google and Facebook's SDKs from the calculation of the k-means clusters to provide us with a lens to investigate the

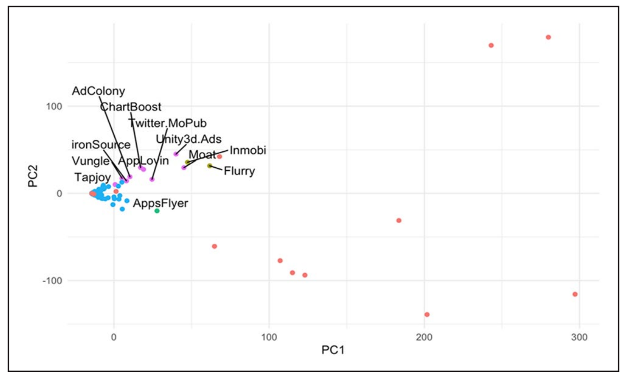

Figure 8. Monetization services power the app ecosystem.

rest of the services ecosystem. Google and Facebook are thus treated as outliers by first creating four clusters of all other SDKs. We then added Google and Facebook as a fifth new SDK cluster to see where they would be placed in the graph in comparison with the other clusters if they were to be treated as new.

The result in Figure 7 puts the clusters of Facebook and Google at the expected distance of the other four clusters with the exception of the already known Google Tag Manager and Facebook Notifications. This confirms their status as outliers from the rest of the SDKs. The visualization also shows what we might miss given the dominance of Google and Facebook. As we ignore them, we can see that the remaining clusters are mainly defined by those services that are associated with Facebook Ad. The next visualization in Figure 8 presents this. It names the services in the three most important clusters that are not from Google or Facebook.

The visualization in Figure 8 underlines how much of the app ecosystem is based on monetization and how much the app internet has become instantiated by programmatic advertising infrastructures, if we consider co-occurrence and technical integration. While Google and Facebook dominate, several other SDKs that provide myriad monetization services continue to grow in importance, seeking out their own niches. This includes, for example, attribution SDKs, which aspire to expand more flexible advertising infrastructures outside of the monopolies established by Facebook and Google. For example, AppsFlyer is one of these attribution SDKs and offers a whole software-as-a-service platform promising brands "a holistic view of every user journey across platforms, channels, and devices."

In Figures 7 and 8, we demonstrated how the technical dependency within the app ecosystem is dominated by monetization interests. To quantify the dominance of Google and Facebook beyond the visualization, we can assign the Google and Facebook SDKs to the remaining clusters to understand how they would be added if we ignored all their existing connections. We calculate the smallest distance that all their SDKs have from the centers of the other clusters. In this way, we determine which k-means clusters they would belong to if they were added newly. We gain a differentiation of the Google and Facebook SDKs and can identify their diverse meanings in the app ecosystem. We see the general-purpose services from Google and Facebook to be close to the main bulk of services. Facebook Login, Google Analytics, Facebook Places, Google Tag Manager, Google Analytics Plugin, and Facebook Notifications all share the co-occurrence characteristics with the main bulk of services. Google DoubleClick, Google Ads, and Facebook Ads are the closest to the small cluster of InMobi and Flurry. They support monetization strategies by specifically focusing on consumer interactions. The remaining Google services, Google Firebase Analytics and Google CrashLytics, are close to the AppsFlyer platform. The remaining Facebook services, Facebook Share and Facebook Analytics, are similarly co-occurring as AppLovin, AdColony, ironSource, and so on. They concentrate on exploiting Facebook data to push monetization. In the digital advertising ecosystem, they gather user data to determine those so-called "consumer touch points" to determine their pre- to post-purchase journey.

Multiplicities of Services

In this section, we consider not just direct co-occurrences but relations between several services within the same app and a neighborhood of apps. We can thus answer questions such as if SDK A co-occurs with SDK B in an app, how is it related to another SDK C that co-occurs with B in another app. To understand such transitions, we use random walk techniques. In a second step, we expand the co-occurrence of two SDKs to multiple SDKs. Here, we employ association rules to zoom in on rising service stars that gather a lot of support from other SDKs—outside the dominance of Google and Facebook.

We can focus on multiple SDK transitions with random walk algorithms, where the SDK bipartite graph is "walked" over several iterations from each of the nodes to create sequences of steps. We are able to map a network such as the bipartite graphs of apps and services so that graph structures

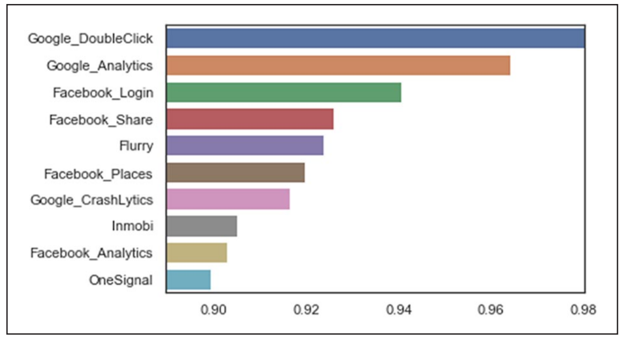

Figure 9. Google Ad DeepWalk Neighborhood.

can be used in machine learning. For instance, starting from Google Ads and randomly walking to a connecting node, we might first go to Smart, then to Adincube, then to Ogury Presage, and so on. If we complete this walk for each node a couple of times, we achieve a very good picture of a node's neighborhood. Afterwards, we continue to do these walks for all nodes to arrive at a complete picture of the graph.

DeepWalk (Perozzi et al., 2014) analyzes such random walk sequences by exploiting again knowledge from computational linguistics and text mining about the vectorization of texts using a neural network technique called Word2Vec. There is a strong analogy between sentences as sequences of words and random walks as sequences of steps between nodes, which allows us to use the advanced knowledge of word embeddings. Using Word2Vec on these sequences of steps, we can represent each node by the community of neighborhood nodes it is connected to as so-called vectors and transitional probabilities between them. Word2Vec offers a powerful way to model network neighborhood transitions that will co-occur and their common context in the whole network.

Using Word2Vec, we can, for instance, zoom in on the relations of particular SDKs. Google Ads is then most similar (most likely to co-occur) to Google DoubleClick, Google Analytics, Facebook Login, and Facebook Share, which confirms some of our earlier results (see Figure 9).

We also gain further differentiation of our earlier results. In Figure 9, Flurry and InMobi are more important than other more generic SDKs from Facebook and Google, which provides us with further evidence that monetization services dominate the ecosystem. OneSignal is new on our list. It is a self-described "world-leader" of push notifications to "create interstitials, banners, and pop-ups that convert" without having to code. It seems to have created a special relationship in the app ecosystem with Google Ads, which leads us to a second layer of powerful services. Those services that can attach themselves to other powerful SDKs such as Google Ads stand out from the rest.

According to DeepWalk, Facebook and Google define the overall distribution of transitions between services for all other apps. Figure 10 is the result of mapping the

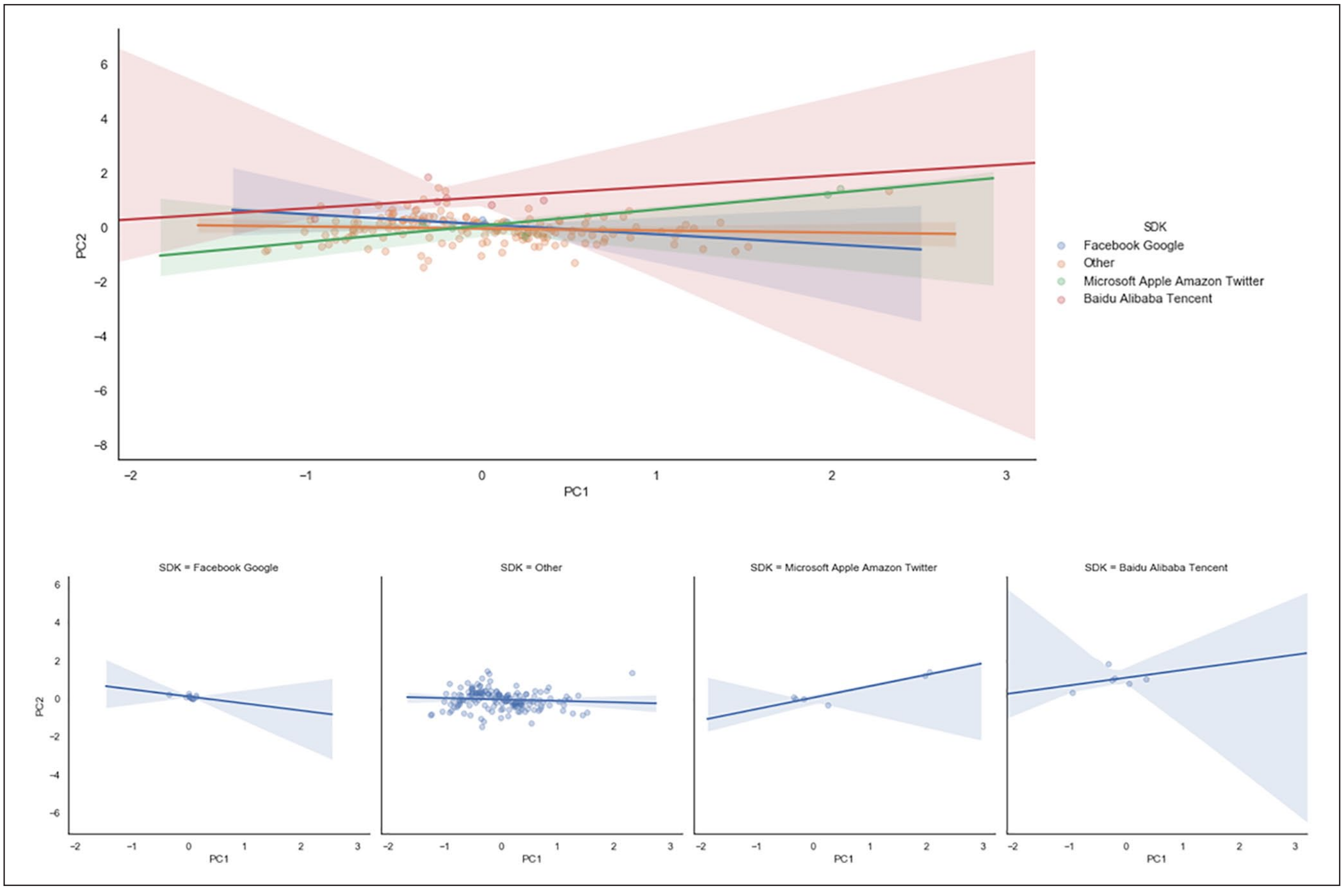

Figure 10. All SDKs are aligned with Facebook and Google.

multidimensional vectors of all services into two dimensions using PCA and then splitting the services into the four groups comparing Facebook/Google and Amazon/Microsoft/Twitter/Apple as well as Baidu/Alibaba/Tencent with the rest. The lines correspond to the linear regression approximating the distributions and demonstrate just how much Facebook and Google determine the spread of service collections over apps once we not only consider two co-occurring collections but their random walk sequences. The regression line of Google and Facebook is very close to the one for the rest of the apps. This means they define a highly similar space and are thus on the same random neighborhood walks. The other US-internet giants follow a different direction with a stronger slope. Really outside are the Chinese giants, which set out a separate space demonstrating their lack of influence on this market, though this might look very different within a China-focused app ecosystem.

DeepWalk, therefore, provides us with the strongest evidence yet of Facebook and Google's dominance by confirming the importance of monetization services. It has offered us insights about a second layer of powerful SDKs, which are competing to instantiate their own infrastructure to access and action personal user data from apps by associating themselves closely with these platforms. It is, however, also the limitation of DeepWalk that it covers the whole graph and thus the overall relations in the ecosystem. We fail to see who the emerging platforms might be. The methods up to now have not allowed us to take a closer look into the bulk of services as Google and Facebook dominated the co-occurrences and transitions. We had to explicitly remove Google and Facebook as outliers to discover more about the co-occurrences of the rest, but that mainly helped us understand their own diverse SDKs' workings. We could also discover for specific services such as Google Ad how new services can attach to them to gain dominance. We are now introducing a new method to decode the rest of the services and find rising stars.

For the networks above, we considered SDKs to be nodes that are related by correlation. The clustering split SDKs into five groups. Both use the idea from text analysis that co-occurrence points to a deeper semantic relation. DeepWalk worked with the metaphor of sequences of nodes as sentences to describe more than just the direct context of services but their transitive neighborhoods. We can expand on this work with an association analysis to identify finding emerging relations. To this end, we employ the Apriori algorithm (Hornik et al., 2005), which find rules such as follows: If we see find an SDK A in an app we also find an SDK B ({SDK A} ⩾{{SDK B}). The Apriori algorithm allows us to associate as many items as necessary into rules but this will

Table 4. Co-occurrences of SDKs by Frequency.

LHS	RHS	Support	Confidence
{AppBrain,Google DoubleClick} ⩾	{Google Ads}	0.014	1.0
{AppBrain,Google Analytics} ⩾	{Google DoubleClick}	0.013	0.97

create a very large set of over 15-m rules. Two example rules are provided in Table 4.

According to the first rule, with 100% confidence we find AppBrain and Google DoubleClick together with Google Ads in an app. Confidence measures how likely it is that where we find the left-hand side (lhs) of the rule we will also see the right-hand side (rhs). The second rule states that AppBrain and Google Analytics imply that Google DoubleClick can also be found (with 97% confidence). Support, on the contrary, provides us with the ability to control the kind of items and rules we would like to concentrate on. Rules with a high support are those that have the most common combinations. Google and Facebook thus have the strongest support, while those rules with a lower support allow us to focus on those relationships that are less common. We could use support therefore to focus on the less common associations, but we will then have an issue to use confidence as a measure of the importance of a rule because we do not know how frequent the consequence relation of the rule is. It could be that it is very frequent, which means that confidence is exaggerated. Instead we will use the lift measure, which is identical to confidence but controls for the frequency of the consequence. Lift compares the frequency of an observed pattern with how often one would expect to see that pattern just by chance (based on its common frequency). A lift close to 1 means that the rule is probably occurring by chance, while the larger the lift the bigger the chance that the rule is not by chance. Concentrating on lift gives us the opportunity to analyze more than the most common services and smaller but important parts of the SDK ecosystem.

To focus on lift, we first prune our larger rule set by removing all rules that are part of larger rules and thus redundant. Furthermore, we are keeping only the rules with the highest lift. Then, the rule with the highest lift is from the Russian-language ecosystem. Where we can see Nexage and Yandex Ad, we can also find Appodeal. Appodeal is overall very popular. Many other SDKs seem to increase the likelihood that we will get Appodeal, too. Appodeal is a US company but has launched a high-profile office in Minsk. It also promises to be a transparent alternative to Facebook and Google, as it is focused on what they call "indie" development (Appodeal, 2020).

Our lift-analysis has discovered Appodeal to be a rising star. The top five rules with the highest lift all include Appodeal, which was launched only in 2015 "made by indies for indies" (Appodeal, 2020). Yandex Ad also appears frequently in the top services' associations by lift, which demonstrates the importance of a growing Russian ecosystem of products and developers. How much Appodeal dominates the lift is shown in Figure 11, where we can see that Appodeal is at the center of the top 25 rules by lift. Items are the labeled vertices, while rules are represented as links between items using arrows.

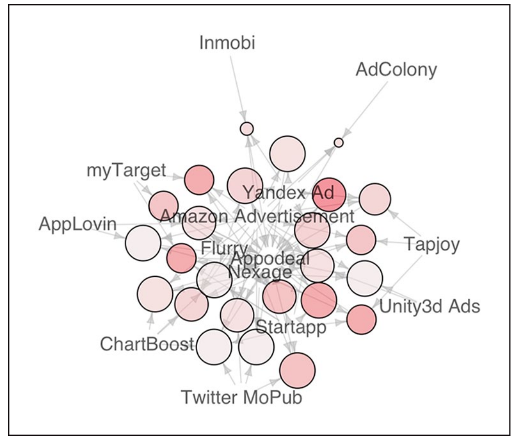

Figure 11. Association network by lift.

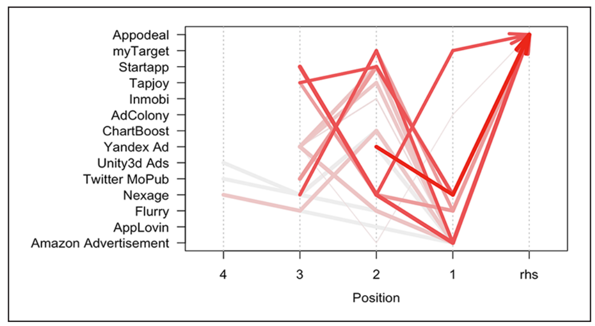

Figure 12. Appodeal's appeal.

A better overview of the relationships and their depth is provided by the coordinate plot in Figure 12, where Appodeal's specific relation to SDKs from the mobile video game world stands out. The figure displays the SDKs on the y-axis, while the x-axis represents the positions in a rule, that is, first item in the rule, second item, and so on. The arrow points to the consequent service. The width of the arrows represents support and the intensity of the color represents confidence. As all arrows are pointing toward Appodeal, it does look to be very popular. All the other gaming SDKs seem to increase the likelihood that a technical relationship with Appodeal develops. But there are also new elements to this new ecosystem of services, for which Amazon becomes more important, as well as more recent marketing services such as InMobi and Flurry.

Conclusion

This article has presented a new perspective in platform studies based on an empirical study of co-occurrence of service collections within apps. Compared with other approaches in platform studies, we have added a perspective on the technical integration of platforms within apps, which makes all platforms depend on each other. The largest platforms dominate here, because they provide the key services for everybody else. Even as we log out of the ecosystems of Google and Facebook, we are still permanently connected to them, as the services they provide via their SDKs reach far into the mobile ecosystem. From the perspective of technical integration, platformization is the permanent process of de- and recomposing SDK services. By de-composing their platforms into service collections the Facebooks and Googles could expand their reach deep into all apps. We have chosen co-occurrence as a methodological tool to find out more about the SDKs all services in the app ecosystem cannot do without.

Two results stand out. First, Google and Facebook dominate this world not just in terms of absolute numbers but also in terms of relative dominance compared with other large internet companies and traditional US competitors as well as newer Chinese ones. They are not just highly present but in terms of co-occurrence they even define the distribution of the neighborhoods for most apps. Google and Facebook also relate most directly to each other. As we rank co-occurrence per frequency, we have to go all the way down to rank 47 to find the first co-occurrence pair that does not include Google or Facebook. Our results have thus shown that Google and Facebook are also heavily technically integrated. Their services commonly co-occur in apps. Beyond their competition they also depend on each other technically.

Second, monetization intentions dominate in the app world. For Facebook, its dominance through mobile SDKs has become so important because as of the third-quarter of 2019, 90% of its advertising revenue came from the mobile ecosystem (Clement, 2020). As we subtracted the dominant Google and Facebook from our SDK co-occurrence considerations, the other stand-out SDKs are seeking to differentiate themselves by providing niche or competing monetization services. These exist outside the expanding closed infrastructures of the dominant platforms. Subsequently, given that SDKs have direct access to the data generated by users on their mobile devices, many new stars are seeking to carve out their own areas of specialization and compete against large infrastructural monopolies. As we controlled for the overall number of appearances, we could see that services like Appodeal seem to have attracted the interest of other SDKs. Over the next few years, this industry will both consolidate and expand, but Appodeal seems to have already worked out a good niche for itself. Our methodology thus enabled us not just to detect existing platform dominations but also rising stars. It will be interesting to observe these developments in the near future.

We have started with the idea to look at platforms from the point of view of technical integration. Compared with the dominant view of them in the critical literature as all-controlling economic monopolies, this view has allowed us to find out how even the largest platforms depend on the technical productions of others. We suspect that this high level of technical integration of the global app ecosystem will provide a stumbling block for any attempts to split the internet, as, for example, the current US administration's attempts to remove Chinese influences. Although we found the reach of Tencent, Alibaba, and so on currently limited, they have left a clear trace in the service assemblages. They will be difficult to remove from the technical integration of apps.

At the same time, platformization through service de- and recomposition also enforces the power of existing monopolies, as a new type of platform power emerges. Those platforms are the most powerful ones that are able to generate services through their SDKs the other digital industries cannot do without anymore, as they depend on them to make profits. There are other powerful service collections, which are those that manage to align themselves with the most powerful ones. Power lies here with those platforms that allow for the growth of the whole ecosystem of apps. These are with Google and Facebook not surprising candidates, but we could also define the scale of their dominance as we compared them with other internet giants. Not all platforms are the same here. On this microlevel of platform power, it is more difficult to change existing associations and relations quickly. Even the Chinese giants seem to still be lacking behind.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: Parts of this work were supported by the Arts and Humanities Research Council (Grant No. AH/R008477/1).

ORCID iD

Tobias Blanke https://orcid.org/0000-0003-0956-987X

References

Anderson, S., & Blanke, T. (2015). Infrastructure as intermeditation–from archives to research infrastructures. *Journal of Documentation*, *71*(6), 1183–1202.

Appodeal. (2020). What's in store for Appodeal in 2020?—Appodeal intelligent ad mediation. https://www.appodeal.com/home/blog/2020-product-roadmap/

Aradau, C., Blanke, T., & Greenway, G. (2019). Acts of digital parasitism: Hacking, humanitarian apps and platformisation. *New Media & Society*, *21*(11–12), 2548–2565.

Baldassarri, D. (2009). Collective action. In P. Hedström, & P. Bearman (Eds.), *Handbook of analytical sociology* (pp. 391–417). Oxford University Press.
Binns, R., Van Kleek, M., Veale, M., Lyngs, U., Zhao, J., & Shadbolt, N. (2018). "It's reducing a human being to a percentage": Perceptions of justice in algorithmic decisions. In *Proceedings of the 2018 CHI conference on human factors in computing systems, CHI '18* (pp. 1–14). Association for Computing Machinery.
Blanke, T. (2014). *Digital asset ecosystems: Rethinking crowds and cloud*. Elsevier.
Blanke, T., & Aradau, C. (2019). Computational genealogy: Continuities and discontinuities in the political rhetoric of US presidents. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*. https://doi.org/10.1080/01615440.2019.1684859
Blanke, T., Greenway, G., Pybus, J., & Cote, M. (2014). Mining mobile youth cultures. In *IEEE international conference on big data (big data)* (pp. 14–17). IEEE Computer Society.
Bratton, B. H. (2015). *The stack*. MIT Press.
Braun, J. (2013). Building the infrastructures of connected viewing. In K. Sanson & J. Holt (Eds.), *Connected viewing: Selling, streaming, & sharing media in the digital age* (pp. 124–143). Routledge.
Burnap, P., Williams, M. L., Sloan, L., Rana, O., Housley, W., Edwards, A., Knight, V., Procter, R., & Voss, A. (2014). Tweeting the terror: Modelling the social media reaction to the Woolwich terrorist attack. *Social Network Analysis and Mining*, *4*(1), 206.
Clement, J. (2020). Facebook: Mobile ad revenue share per quarter 2019. *Statista*. https://www.statista.com/statistics/999580/share-of-mobile-facebook-ad-revenue-quarter/
Coté, M., & Pybus, J. (2016). 'Simondon on datafication. A techno-cultural method'. *Digital Culture & Society*, 2(2), 75–91. https://doi.org/10.14361/dcs-2016-0206
Dance, G. J. X., LaForgia, M., & Confessore, N. (2018). As Facebook raised a privacy wall, it carved an opening for tech giants. *The New York Times*. https://www.nytimes.com/2018/12/18/technology/facebook-privacy.html
Datadog. (2018). Airbnb's journey to a service-oriented architecture at 1,000-engineer scale. https://www.datadoghq.com/videos/airbnbs-journey-to-service-oriented-architecture-at-1000-engineer-scale/
Exodus Privacy. (2020). Exodus privacy. https://exodus-privacy.eu.org/en/
Fuhse, J. A. (2015). Theorizing social networks: The relational sociology of and around Harrison white. *International Review of Sociology*, *25*(1), 15–44. https://doi.org/10.1080/03906701.2014.997968
Gillespie, T. (2018). *Custodians of the internet: Platforms, content moderation, and the hidden decisions that shape social media*. Yale University Press.
Harris, Z. S. (1957). Co-occurrence and transformation in linguistic structure. *Language*, *33*(3), 283–340.
Hellsten, I., & Leydesdorff, L. (2020). Automated analysis of actor–topic networks on twitter: New approaches to the analysis of socio-semantic networks. *Journal of the Association for Information Science and Technology*, *71*(1), 3–15.
Helmond, A. (2015). The platformization of the web: Making web data platform ready. *Social Media+ Society*, *1*(2), 2056305115603080.
Helmond, A., Nieborg, D. B., & van der Vlist, F. N. (2019). Facebook's evolution: Development of a platform-as-infrastructure. *Internet Histories*, *3*(2), 123–146.
Hornik, K., Grün, B., & Hahsler, M. (2005). Arules-a computational environment for mining association rules and frequent item sets. *Journal of Statistical Software*, *14*(15), 1–25.
Jiang, L., & Yang, C. C. (2013). Using co-occurrence analysis to expand consumer health vocabularies from social media data. In *IEEE international conference on healthcare informatics* (pp. 74–81). IEEE.
McStay, A. (2018). *Emotional AI: The rise of empathic media*. SAGE.
Nieborg, D. B., & Poell, T. (2018). The platformization of cultural production: Theorizing the contingent cultural commodity. *New Media & Society*, *20*(11), 4275–4292.
Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). DeepWalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '14* (pp. 701–710). Association for Computing Machinery.
Plantin, J.-C., Lagoze, C., Edwards, P. N., & Sandvig, C. (2018). Infrastructure studies meet platform studies in the age of Google and Facebook. *New Media & Society*, *20*(1), 293–310.
Protalinski, E. (2019). Amazon reports $70.0 billion in Q3 2019 revenue: AWS up 35%, subscriptions up 34%, and "other" up 44%. *VentureBeat*. https://venturebeat.com/2019/10/24/amazon-earnings-q3-2019/
Pybus, J., Coté, M., & Blanke, T. (2015). Hacking the social life of big data. *Big Data & Society*, *2*(2), 2053951715616649.
Rieder, B., & Sire, G. (2014). Conflicts of interest and incentives to bias: A microeconomic critique of Google's tangled position on the web. *New Media & Society*, *16*(2), 195–211.
Srnicek, N. (2017). *Platform capitalism*. John Wiley & Sons.
Van Dijck, J., Poell, T., & De Waal, M. (2018). *The platform society: Public values in a connective world*. Oxford University Press.
Zuboff, S. (2019). *The age of surveillance capitalism: The fight for a human future at the new frontier of power*. Profile Books.

Author Biographies

Tobias Blanke is University Professor of Artificial Intelligence and Humanities at the University of Amsterdam. His academic background is in moral philosophy and computer science. His principal research interests lie in the development and research of artificial intelligence (AI) and big data devices as well as infrastructures for research, particularly in the human sciences. Recently, he has also extensively published on ethical questions of AI like predictive policing or algorithmic otherings, as well as critical digital practices and the engagement with digital platforms. He has authored 3 books, over 80 journal articles, conference papers, and book chapters spanning traditional philosophy, computer science, and digital studies. His monographs include, most recently, *Digital Asset Ecosystems—Rethinking Crowds and Clouds*, which offers a new perspective on the collaboration between humans and computers in global digital workflows. He is currently writing a book on the socio-economic position of AI called *Algorithmic Reason—the Governance of Self and Other*.

Jennifer Pybus is Lecturer in Digital Culture and Society at King's College London. She is a globally recognized, interdisciplinary, early-career scholar with an international reputation around processes of datafication, privacy, and data literacy. Her research investigates those critical points of tension that lie at the intersection of digital culture, AI and algorithmic processes, and emerging infrastructures that are set up to capture and action personal data. Part of this work focuses on the political economy of social media platforms, display ad economies, and the rise of third parties embedded in the mobile ecosystem. She is currently writing a book that critically explores the socio-cultural and political–economic dimensions of technical objects, which facilitate processes of datafication embodied in the production, circulations, and economic data flows, entitled: *iData: Politics of the Personalised Economy*.