Sage Journals

Social Media + Society

Impact Factor: **4.9**
5-Year Impact Factor: **6.8**

Contents | PDF/EPUB | More

Abstract

Researchers, policymakers, and competition and regulation authorities worldwide recognize the utility of application programming interfaces (APIs) in powering the digital economy and driving datafication and platformization processes. However, it remains unclear how the APIs of leading social media relate to platform governance and how this relationship evolved. This article traces the evolution of Facebook's APIs, which evolved from a relatively simple programming interface for data access into a complex layered and interconnected *governance arrangement*. The study draws on a large corpus of (archived) developer pages and API reference documentation to examine the history of Facebook's API governance; that is, the governance of and by Facebook through its APIs. This historical analysis emphasizes the technical dimensions and dynamics of what, how, and whom powerful platforms seek to govern, thus highlighting the technicity of platform governance and how it evolved. Because APIs facilitate and govern the material conditions of app development and the social and economic processes they sustain, powerful platforms influence the evolution of their larger ecosystems. As such, the technicity of Facebook's API governance represents a major source of the platform's "infrastructural power."

Introduction

Researchers, policymakers, and competition and regulation authorities worldwide recognize application programming interfaces (APIs) for their role in datafication and platformization processes as a way to "dominate the digital world" (FT Reporters, 2020; Iyer & Getchell, 2018; Van Dijck, 2021). APIs serve as the *lingua franca* for the exchange of data and services between companies and are of strategic importance for platform companies like Google and

Facebook. This is in part due to the web becoming more data-intensive with the rise of the platform as its dominant technological and business model (Helmond, 2015). APIs have become the core elements of digital infrastructure, underpinning today's platform economy and society. Consequently, Iyer and Getchell (2018) warn that regulators should not only focus on the market dominance of platform companies but also on their "data dominance"—specifically, how platform companies use APIs to share data or integrate their services with third parties. Competition authorities and regulators in Europe and the United States increasingly scrutinize anti-competitive uses and potential misuses of data centered around Facebook's APIs and the platform's monopoly power (CMA, 2020: Appendix J; Federal Trade Commission [FTC], 2020). The FTC (2019a) fined Facebook US$5 billion for violating consumer privacy rights by providing third-party developers access to users' friends data via its APIs and for not properly reviewing third-party developers and their apps. The FTC also ruled that app developer Aleksandr Kogan and Cambridge Analytica CEO Alexander Nix used deceptive data-gathering practices to harvest personal information from Facebook users and their friends through a quiz app that exploited the Facebook Graph API (GAPI) (Albright, 2018; FTC, 2019b). These accounts suggest the need for a better understanding of the role of APIs in contemporary platform society.

APIs enable programmatic communication and the exchange of data and functionalities between software systems, such as digital platforms. They power the platform economy and serve as the core infrastructural elements that underpin the large ecosystems of apps and services (or "complements") created by third parties and partners (or "complementors"). APIs play a key role in the capture and movement of personal data, the interconnections between software apps and services, and bring about "the formation of platform monopolies" through the decentralization of their services (Blanke & Pybus, 2020). Although APIs may be perceived as "microscopic" technical objects, they are nonetheless significant because they comprise the material infrastructures of platforms and apps and articulate and shape the processes of datafication (e.g., programmatic data-sharing) and platformization (Helmond, 2015; Pybus & Coté, 2021). APIs are more than technical objects designed and deployed by platforms at a given moment; they have become complex layered and interconnected technical objects that govern a platform's data and services and are prone to constant change due to both internal and external pressures. Particularly, we find that APIs evolve through the interactivity between a platform and its communities of use, app development, and monetization by businesses, as well as through pressures originating from a platform's competitive and regulatory environments.

Despite broad recognition of their importance in the platform society, there is no comprehensive understanding of social media platforms' APIs as complex technical objects that change and evolve continuously. As such, critical scholars argue that "[r]egulatory fixes require detailed insights into how technology and business models work" (van Dijck et al., 2018, p. 158) and call for the "observability" of platforms as an explicit means of regulation (Rieder & Hofmann, 2020). Furthermore, very little is known about the relationality between platforms' APIs, governance, and power. Platform companies design and change their APIs to facilitate third-party app development in ways that influence (or "orchestrate") the evolution of their ecosystems (Tiwana et al., 2010). These ecosystems typically comprise multiple user and stakeholder groups connected to the same core technical platform using one or more of its APIs, including (small, medium-sized, and large) app development ventures, businesses, digital marketers and advertisers, and academic researchers. In today's platform society, APIs enable and control the possible relationships and interactions between these different users and stakeholders and thus serve as a core technical dimension of "platform governance" (Gorwa, 2019, p. 854). Accordingly, platform companies like Facebook and Twitter have responded to public controversy, criticism, and external social pressures not only with feature and policy changes (Barrett & Kreiss, 2019) but also with amendments to their APIs. After the Facebook–Cambridge Analytica (Fb–CA) data scandal, Facebook made changes to certain terms and policies and implemented restrictions on API data access and sharing.

This article presents the results of an empirical case study of the structure and evolution of Facebook's APIs and their relation to platform governance to highlight the *technicity*—that is, the technical dimension and dynamics—of what, how, and whom platforms like Facebook seek to govern. The analysis is focused on Facebook's APIs, which have been among the most popular, widely used, and most controversial for over a decade (Albright, 2018). Specifically, we consider the coevolution of Facebook Platform and its APIs and modes of platform governance on three levels: (1) the structure of Facebook's entire API architecture; (2) core API objects in terms of their properties, connections, and parameters; and (3) their associated permissions, as handled through Facebook applications and Facebook Login specifically, using both current and archived developer pages. We provide original empirical materials for further historical platform research to better understand the evolutionary dynamics between API design and platform governance (e.g., Helmond et al., 2019; Helmond & van der Vlist, 2019; Nieborg & Helmond, 2019).

Building on prior research, the analysis traces how Facebook's APIs have evolved from a simple programming interface for development into a complex layered and interconnected *governance arrangement* wherein technical API specifications serve to enforce (changes to) platform policy and strategy. We thus contend that platform governance is about more than a platform's content moderation, terms and policies, or corporate governance structure; it also involves the (re)design of technical API specifications that govern and control the possibilities for the exchange of data and services between software systems and organizations. As such, this study makes an empirical–historical contribution to the ongoing debate on "platform governance" within the literature on social media and platform studies (e.g., Caplan & Gillespie, 2020; Gillespie, 2018; Gorwa, 2019; Gorwa et al., 2020; Medzini, 2021; Schreieck et al., 2018). We focus specifically on the technicity of Facebook's platform governance as enforced through APIs, which, as scholars note, "are an important source of infrastructural power in the platform society" (Busch, 2021; cf. Blanke & Pybus, 2020; Munn, 2020). In fact, "infrastructural power" is one of the identified "blindspots" in the European policy debate on "platform power" (Busch et al., 2021). Our contribution emphasizes that APIs are neither simple nor stable technical objects but complex governance arrangements that continuously change and evolve, which is crucial to gaining an understanding of how APIs tie into platforms' power.

In the next section, we first provide an overview of the current literature on APIs and platform governance to position our contribution in these research areas. Second, we detail our historical approach to examining Facebook's API governance against this background and describe our method of data collection and openly available data set. Third, we present our empirical case study of the structure and evolution of Facebook's APIs. Finally, we discuss the implications of the historical analysis, which highlights the importance of technical perspectives on the governance of and by APIs as a major source of platforms' infrastructural power within the ecosystem.

API Studies Meet Platform Governance

APIs have been studied by scholars across multiple disciplines and fields, including media and communication studies, information systems (IS) research, and software engineering. This section identifies relevant streams of research on the relationality between APIs and governance to contextualize our analysis of the technicity of platform governance.

Most importantly, APIs have been described as mechanisms of generativity and infrastructural control. APIs enable third-party app developers to interact with a platform to access and exchange data and services through a standardized information exchange that ensures interoperability (Bodle, 2011; Tiwana, 2014, p. 7). They coordinate development work between platforms and third parties (cf. de Souza et al., 2004), which means that platform governance through APIs is also a practical matter of facilitating collaboration. Platforms stimulate generativity by inviting third-party app developers to create new apps and services "on top" of a platform by using its APIs. This generative dimension of platforms has previously been understood as a form of participatory "remix" or "mashup" culture, as platform appropriation or reimagination, and as a value-adding activity (Bucher, 2013; Evans & Basole, 2016; Gerlitz et al., 2019; Hogan, 2018; Werning, 2017). At the same time, there is a "paradoxical relationship" between generativity and control because platform owners must maintain economic, social, and technical control over their platforms, the external contributions of third-party app developers, and the platform's evolution (Eaton et al., 2015; Ghazawneh & Henfridsson, 2012). As such, APIs provide access to data and services in exchange for infrastructural control (Eaton et al., 2015; Evans & Basole, 2016). Protocological technical objects like APIs serve as "conduits for governance"—or as "artifacts of governance"—where control is enabled on the level of API code (Bucher, 2013; Musiani, 2013; Snodgrass & Soon, 2019). Control over APIs "amounts to control over the platform and its evolution" and the platform's complements and complementors (Tiwana et al., 2010, p. 680). APIs thus facilitate infrastructural dependencies between platforms and apps, which we argue represents a source of infrastructural power and provides control over the platform's ecosystem.

Second, the rise in popularity of (proprietary) APIs over open web development standards to enable generative practices has transformed the fabric of the open web and beyond. As web APIs and social plugins began to promote a more "social" experience of the web in the early 2010s, new forms of API-based connectivity emerged to underpin today's "data-intensive" web (Gerlitz & Helmond, 2013). As Langlois and Elmer (2019) suggest, platforms are increasingly "weaving themselves in a new distributed infrastructure of life in all its forms" (p. 6) in which APIs enable new "connected viewing environments" in the television industry (Lahey, 2016) and new "data seams" in the urban fabric of cities (Raetzsch et al., 2019). These developments are centered on the role of APIs as the standard mechanism for interconnectivity, embeddedness, and scale growth. However, they also raise concerns [illegible]nd power through platformization (Blanke & Pybus, 2020; van Dijck, 2021) and

"infrastructuralisation," whereby platform-based services acquire the characteristics of infrastructure (Helmond et al., 2019; Plantin et al., 2018). The outcome is that these new data-intensive fabrics are no longer open or public but are instead privatized and governed by platform companies.

Third, APIs structure and "datafy" social and commercial processes. Social media platforms use APIs to create and temporarily stabilize digital identities for consumption by external apps (Pridmore, 2016). On the consumer side of their platforms, social media companies like Facebook are infrastructuring online sociality for the eventual monetization of targetable audiences (Alaimo & Kallinikos, 2019). On the developer side of these platforms, webmasters and developers implement APIs to ensure the seamless integration of their content and pages by making them "platform-ready" (Helmond, 2015). However, as social media companies seek to create an "advertiser-friendly atmosphere of connectivity," their APIs are "largely blind to acts of disconnectivity, such as unfriending and unliking," thereby only datafying commercially relevant types of sociality (John & Nissenbaum, 2018).

Fourth, APIs have been pivotal to the business models and strategies of platforms and the commercialization of the internet in general. IS researchers have studied the economic and business dimensions of the API ecosystem as an API "economy." In such an economy, platform companies strategically provide data access through APIs to stimulate the development of API "mashups" to capture the value produced by these complements (Evans & Basole, 2016). APIs also facilitate the distributed capture of datafied user engagements on third-party websites and apps, giving rise to web economies, such as the "like economy" of the social web (Gerlitz & Helmond, 2013). The logic of participation thus became heavily commercialized. Some researchers have traced TripAdvisor's evolution into a diversified ecosystem of data-based services wherein complementors assemble around new data forms to create additional services governed through APIs (Alaimo et al., 2020); others have mapped extensive API-based business-to-business (data) partnership networks that integrate social media platforms within the global digital advertising market (van der Vlist & Helmond, 2021). Furthermore, the strategic role of APIs in collecting new types of valuable data has been studied to understand the evolution of data and the business strategies of platforms (Bechmann, 2013; Wilken, 2014) by comparing API ecosystems (Evans & Basole, 2016). In short, APIs are not merely technical objects for software and app development but also an integral part of a platform's data and business strategy.

Finally, APIs are commonly used and reflected upon as tools for academic research. Some researchers have examined the use of APIs for data extraction purposes and the role of API-

based research software tools as “data makers” (Rieder, 2013; Vis, 2013). Similarly, others have considered the technicity of APIs as they intervene in empirical research by shaping their objects or phenomena of study (e.g., Rieder et al., 2015). As far as Facebook is concerned, research uses are just another app type: they use the same APIs as other third-party app developers but for different purposes.[1] However, even minor API changes can have significant research implications. Newly imposed data limitations may introduce potential biases that undermine the representativeness of data studies (Ho, 2020). Such API-based studies can arguably only be interpreted and replicated alongside historical information about how the APIs used have changed and evolved. After the Fb–CA data scandal and the subsequent “APIcalypse” (Bruns, 2019), platforms like Facebook and Twitter severely restricted their API data access and sharing practices. This impacted critical academic research into phenomena, such as online abuse, hate speech, and disinformation campaigns that employed APIs for their data collection. As a result, the conditions of platform observability through APIs have worsened (Rieder & Hofmann, 2020). How platform APIs are governed raises questions around fair use and the need to look for alternative research methods suitable for a “post-API environment” (Perriam et al., 2020; Venturini & Rogers, 2019).

While these streams of API-related research all provide important insights into the politics of social media platform APIs, they have not necessarily focused on the evolutionary dynamics of APIs and their complicated role in how platforms govern app and business development. On the one hand, media and communication scholars have emphasized governance concerning consumers and content, including platforms’ policies and their terms and conditions (e.g., Gillespie, 2018), the technical challenges and politics of algorithmic content moderation (Gorwa et al., 2020), and governance by algorithms (Musiani, 2013). To this end, platforms enact governance by making certain design choices in their technical infrastructures, app features, and other architectural elements (DeNardis & Hackl, 2015; Duguay et al., 2020). On the other hand, IS researchers and software engineers have conducted empirical studies on APIs and their documentation to understand structural platform changes and how they are communicated and impact development (e.g., Medjaoui et al., 2018; Sohan et al., 2015). They have also theorized how a platform’s evolution is influenced by the coevolution of its architecture, governance, the “environmental dynamics” of its ecosystem (Tiwana et al., 2010), and the challenges of governing platform ecosystems (Schreieck et al., 2018). In studying the technicity of platform governance, we draw from each of these strands. Governance of and by APIs concerns a platform’s developer and business

communities (and indirectly, its consumers), and involves its terms and policies, API design, and strategy.

Furthermore, our historical approach foregrounds the ability of a platform to shape—through governance and strategy—the evolution of the ecosystem around it in ways that impact certain outcomes, social, cultural, economic, or otherwise. For instance, these may include forms of "platform envelopment" (or "capture"), where a platform owner leverages power asymmetries over dependents to move into another's market (e.g., Partin, 2020), or "path dependency" and proprietary "lock-in" effects, where a platform benefits from continued use by consumers, developers, or businesses based on historical preference or specific use (e.g., Alaimo et al., 2020). Previous technical design choices, strategic decisions, and advantages thus often sustain an enduring influence on the present.

Studying API Evolution and Governance

Platforms, such as Facebook leave many forms of material traces that document their operations; these can subsequently be used to observe platforms (Rieder & Hofmann, 2020) and reconstruct a platform's means of governance and strategy throughout their evolution. These material traces include information about the platform's APIs and other software (development) tools located on dedicated websites for developers (developers.facebook.com) and businesses (facebook.com/business). Because these websites have been well-archived in online archives, they are particularly suitable for empirical and historical platform studies (Helmond & van der Vlist, 2019).

Just as APIs are not stable objects but are prone to continuous changes, so does their imagined utility for third-party development evolve. Moreover, platform owners, complementors, and other actors collectively shape and reshape the evolution of a platform's boundary resources in a process of "distributed tuning," revealing the dualistic logic of generativity and infrastructural control (Eaton et al., 2015; Gerlitz et al., 2019). In this process, a platform's reference documentation serves both a functional and strategic role by "optimizing the developer experience" in working with the API (Medjaoui et al., 2018) while acting as "a conspicuous form of political communication" that enacts "specific social roles" (Moschini & Sindoni, 2021); for example, in shaping the meaning of "privacy" (Greene & Shilton, 2017). They provide important information and contain traces of API governance—that is, governance of and by platforms through their APIs. The amount of detail provided in API reference documentation thus enables granular empirical analysis of the material

conditions of third-party app development and how those conditions have changed and evolved. These material conditions shape data flows and the kinds of interactions that are supported by a platform, which enable and constrain development and business opportunities. We provide detailed empirical evidence to learn how API governance operates and how it evolved over the years.

The Facebook for Developers website covers the reference documentation for Facebook Platform.[2] This includes technical information about each of its open (i.e., public-facing) APIs, instructions on how to use them (e.g., how to read or write information to Facebook), and additional information about versioning, access levels, and rate limits, as well as specific data fields, edges, parameters, and permissions (specifying which data and functionality can be accessed under which requirements). Within the API reference documentation, Facebook currently refers to API objects, which represent information on Facebook, as "nodes" or "endpoints" (e.g., the /user, /photo, /event, and /page nodes). The properties associated with a node are "fields" (e.g., "name" and "birthday" are fields of the /user node) and some fields require permissions from the user (e.g., the "location" field). Connections between the nodes are referred to as "edges," e.g., /(user-id)/feed returns any posts and links shared by a specified user-id on their profile. Although there are many distinctly named APIs, they can mostly be accessed through the same base URL (i.e., graph.facebook.com). These "nodes" and "edges" thus correspond with what consumers can see and do on Facebook's "front-end" consumer interface, although the API (programming interface) offers different affordances (and sometimes more detailed information than the consumer interface) for development purposes.

Developers can use these nodes and edges ("endpoints") to access Facebook data and services. When developers request data from a node (e.g., /user), it typically returns not one but many data points about that node (i.e., its properties, such as "birthday" or "gender"). Furthermore, when they connect to an edge, they can retrieve all the nodes associated with that edge. Since the beta launch in 2006, a complex layered structure of access controls, application permissions, and app review guidelines has evolved to govern and restrict API data access for most nodes and edges. Finally, we make use of Facebook's archived Platform Terms and Developer Policies, which explain how the Platform should (and should not) be used (FD-2021a; FD-2021o).

Archived Developer Pages and API Reference Documentation

The empirical analysis is based on Facebook's developer pages as retrieved from the "live" web and the Internet Archive. We downloaded 3,394 "live" web pages from developers.facebook.com (2019–2020) and retrieved 1,960,901 developer pages from the Wayback Machine, going back to the initial launch of Facebook's beta API (August 2006–February 2020).[3] Because Facebook does not provide an archive of its developer website, these independently archived sources provide an important means to analyze the observability of the platform. We combined multiple strategies to explore this large corpus of web sources because APIs are complex composite technical objects that demand analysis at the different levels at which they occur and operate. Therefore, we analyze the evolution of Facebook's APIs on the level of the entire API architecture, of individual API objects, and application permissions.

On the larger level of the API architecture, we derived the link structure of 63,027 reference documentation pages that describe Facebook's APIs. Each page describes a specific node and any associated fields, edges, and parameters, and details the data or functionality that is available to third-party developers. As such, the link structure embedded in the reference documentation reflects the API architecture. We derived and charted the link structure as it evolved with each new version. In addition, we created a corpus of 178,972 web pages with annual "snapshots" of archived URLs anywhere on the Facebook for Developers site to visualize the complexity and diversity of the APIs and to examine their naming conventions. We further analyzed the associated "changelogs" (FD-2021c; FD-2021g), which document all versioned API changes and include information about newly introduced, changed, and deprecated nodes and edges as well as information about permission changes. A changelog addresses third-party app developers and communicates implemented or planned API changes and their implications. In some cases, they also reveal how Facebook Platform has responded to public controversy and external social pressures or made changes to its business and data strategies.

On the level of individual API objects, we examined one of the core (and most connective) nodes in the entire reference documentation: the GAPI User object (FD-2021i). The User object represents a user on Facebook (i.e., an account that represents a person). As such, the object is central to the API because it is at the core of Facebook's social network, which is structured around people's user-profiles and friendship connections, and its advertising-based revenue model, which lets paying customers find and reach those users with targeted messages. We thus reconstructed how the User object evolved as a data object (in terms of descriptive properties) and also as a *relational data object* (in terms of its connections). The

User and Page objects—the latter representing businesses, organizations, and public figures—are the two nodes that can authorizes API access tokens for apps to allow data access. Other data objects are typically linked through the User object in some way. We further examined the evolution of targeting options for finding and reaching Facebook users with the marketing API (MAPI), which is a distinct subcomponent of the GAPI used by Facebook Marketing Partners to develop marketing and advertising tools and services. The same targeting options are available through Facebook's self-service advertising tools, enabling us to examine how the targetable user has been governed through the MAPI (FB-2021).

Finally, we examined application permissions; these provide a way for apps to access data from Facebook (FD-2021l) and have become an increasingly important governance mechanism. We examined the structure of these permissions and when specific nodes or edges require permission from the user in the first place. Facebook currently distinguishes between "Basic" and "Extended" Permissions for accessing its data and services. Any app can, by default, access ("read") the data fields that belong to a User's "public_profile," including their "name" and "picture." When an app requires access to additional data or to publish ("write") data to the platform, it needs to request extended permissions from the respective user(s).

To contextualize the observed changes on all levels, we consulted Facebook's own Developer Blog and News sections, as well as external technology journalism blogs, interviews and testimonies by CEO Mark Zuckerberg, and 7,000 pages of documents leaked during Facebook's litigation with app developer Six4Three in court (Appendix; FL-2019a). We thus used a variety of primary (i.e., Facebook) sources and external sources that provide important contextual information about specific changes. Finally, we used visualizations to support the analysis and communicate a sense of the complexity of Facebook's APIs and made our data sets openly available.

The Structure and Evolution of Facebook's APIs on Three Levels

This section presents the results of a multilevel analysis of the structure and evolution of Facebook's APIs in terms of what, how, or whom it governed; that is, we trace the emergence of Facebook's API governance arrangement beyond its mere programming interface across three levels. To begin with, the architecture level reveals how changes to Facebook's core APIs were central to the platform's evolution and how they became objects of governance. In addition, the object level shows how Facebook represents its users as data objects,

determining what counts as user data and how it may be accessed. Finally, the permissions level shows how Facebook articulates more granular access controls organized as an additional layer atop the API. Taken together, they provide important insights into how changes made to Facebook's APIs relate to the platform's governance and (data) strategy, especially regarding the orchestration of (asymmetric) relationships with complementors in the ecosystem (cf. Tiwana et al., 2010).

API Architecture Level

At the API architecture level, Facebook Platform has evolved from a single programming interface into a web of interrelated API components; that is, collections of API endpoints around core platform products (e.g., Messenger, Instagram Platform's APIs, etc.). Initially (2006–2010), the platform only included the Facebook API, which provided data access to Facebook's core platform products (e.g., Profile, Friends, Photos, and Events). This enabled developers "to add social context" to their Facebook apps (FD-2007). This "RESTful" API grew in size with the addition of further API functionality,[4] reflecting Facebook's evolution as a social network. With the launch of the current GAPI (2010), this API architecture style was redesigned on the logic of the (social) graph, which modeled Facebook's social network entirely in terms of "nodes" (objects) and "edges" (connections). Since then, the graph model has codified and represented ("datafied") any kind of relationship between people and those between people, objects, and activities both on and off the platform (FD-2021h).

Core API Components: Diversification and Integration

Since 2010, many new core APIs have been introduced and integrated, reflecting the different tools, products, and services that Facebook has created (e.g., the MAPI, Messenger Platform, Workplace, etc.) or acquired (e.g., Instagram, WhatsApp, Atlas, etc.) over the years. Under each of these core API components of Facebook Platform, we find more specific API endpoints that provide access to individual data objects and functionality components. Figure 1 presents the evolution of the entirety of the architecture of Facebook's APIs, including core components and specific endpoints as well as their interrelations. This API architecture grew not only in size and complexity but also became increasingly interconnected as Facebook evolved from a social network into a multi-sided platform for development, underpinning a large ecosystem of data-based apps and services (Alaimo et al., 2020; Helmond et al., 2019). Here, governance manifests in the strategic diversification and integration of Facebook's APIs and the ongoing (re)design and (re)structuring of API components. It also manifests in tiered API access levels and rate limits, which control API

access for different users and limit the number of API (data) requests that can be made within a given time.

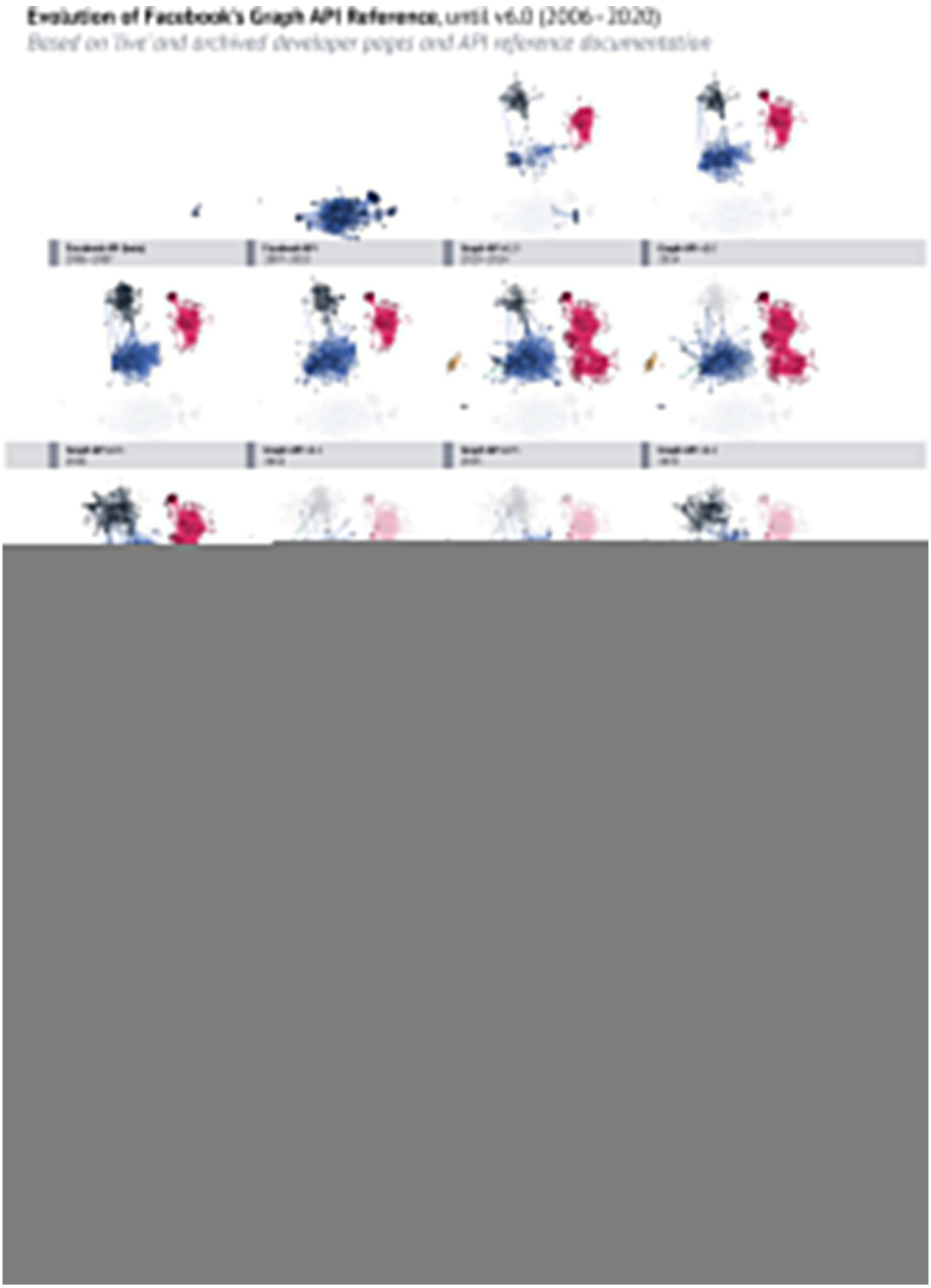


Figure 1. Evolution of Facebook's API Reference until v6.0 (2006–2020) (small multiples). Each tile represents the entirety of the API architecture for one API version (i.e., accumulated link structure of

the entire API reference documentation), while the “legacy” architectures from all previous versions are rendered transparent in the background.

Since 2014, the MAPI has increasingly merged (or technically integrated) with the GAPI. As such, Facebook’s advertising platform became part of its core development platform rather than remaining separate. While Facebook’s own self-built and acquired platforms—Facebook’s “family of apps”—have remained separate for consumers, we see that they have long been interconnected in the back-end for business users (Nieborg & Helmond, 2019). These changes were slowly rolled out throughout several API versions. When Instagram (2012) and WhatsApp (2014) were acquired, their back-ends initially remained entirely separate from the rest of Facebook Platform. They later migrated to Facebook’s data centers to “ease the integration with other internal Facebook systems” by unifying “their underlying technical infrastructure” to further “increase Facebook’s utility and keep users highly engaged inside the company’s ecosystem” (IE-2014; Isaac, 2019).

The longer-term evolutionary trajectory of Facebook’s API architecture is characterized by stages of explosive growth (a diversification of API endpoints) combined with an ongoing integration process of the platform’s core API components. These changes are as much discursive as they are technical: we find a proliferation of distinct APIs mentioned and documented in the API reference documentation. “The Facebook API,” as it was originally called, gradually evolved into a complex and interrelated structure comprising hundreds of distinct APIs to address more specific development needs.

Figure 2 lists all the entities referred to by Facebook as “APIs” between 2006 and 2019. We identified 446 unique APIs in total, comprising both individual and collections of endpoints. The vast majority encapsulate very specific GAPI and MAPI data, and services that enable developers to build tools, products, and services more securely and efficiently. These special-purpose APIs can be “in name only,” merely (re)packaging API functionality to promote particular use cases and enable targeted and subtle forms of governance. As such, the developer pages do not only contain technical information but also have a communicative function for developers by signaling specific use cases (cf. Dal Bianco et al., 2014). Because of this dual role, we see Facebook’s evolution reflected in the reference documentation and in how its API architecture is presented and described.

Figure 2. Evolution and overview of "API" mentions within Facebook's reference documentation, 2006–2019.

Changelog: Toward a Stable Platform

Developers require stable platforms on which to build and maintain their apps and services. Indeed, any dependent apps and services would immediately break without timely and clear communication and instructions (documentation) about upcoming API changes (Sohan et al., 2015). Reversely, the platform owner risks losing its integrations and embeddings in other competitive industries and societal domains.

At first, Facebook did not systematically communicate its API changes and only documented some of them on its Developer News page (FD-2006a). This is reflected in Facebook's internal motto at the time to "move fast and break things," which impacted dependent app developers. Consequently, due to mounting criticism, Facebook started publishing a Developer Roadmap in 2010 to "help developers plan for changes" (FD-2010d). This roadmap was part of Facebook's "Operation Developer Love" (2010–2011), which was intended to ease tensions with developers requesting improved communication about changes to the platform to increase reliability (FD-2010a). Before this, Facebook would primarily use its Developer Terms of Service to govern app development. For instance, the increasing popularity of "social games" built using Facebook's APIs led to the so-called "News Feed spam wars" as developers tapped into the possibilities of Facebook's APIs to virally distribute their apps and games (Levy, 2020, p. 165). After user complaints about these "spammy" game apps, Facebook immediately restricted developers' API access to the News Feed and notifications, citing their terms of service. This event would kickstart a "push-pull between Facebook and its developers," where "Facebook would change the rules and developers would figure out how to get around those rules" (Levy, 2020, p. 165).

The operation eventually led to stability improvements when the reference documentation became more frequently updated and a "breaking change policy," signaling API changes that would cause an app to malfunction in advance, was introduced (FD-2011a; FD-2011b). Facebook later started posting weekly updates on upcoming platform changes on its Developer Blog (formerly Developer News), marking its transition toward becoming a more stable platform (2010–2014) (FD-2010a). This transition from an experimental to a stable development platform has been critical in Facebook's acquired infrastructural scope and

scale (Helmond et al., 2019) because it reduced or minimized development risks for complementors, particularly businesses.

With the release of GAPI v2.0 (2014), Facebook made several key changes to announce and document API changes: it introduced "versioning" (and retrospective version numbers) to manage the multiple (consecutive or parallel) releases of an API as well as to communicate upcoming changes, provide a transition period to provide ample time to address them (to avoid "breaking"), and make a stability guarantee for developers. Moreover, Facebook introduced the changelog to announce and document any changes to the GAPI (FD-2014a). Concurrently, Facebook changed its internal motto to "move fast with stable infrastructure" (Levy, 2020, p. 243). The platform further introduced versioning to the Ads API (now MAPI) and aligned its versioning and release cycles with the core GAPI soon after (FD-2014b).

The shift Facebook made from its continuous trajectory of development and releases (with concomitant unpredictability) to scheduled and versioned release cycles mark an important step in the platform's evolution. Communication between Facebook and third-party developers was further standardized through developer pages and reference documentation. Since then, changes to the GAPI and MAPI have been documented in the changelog because of its important communicative function, namely, that it informs developers (and other complementors) about whether, how, and when they should update their tools, products, and services to comply with a new API version (along with an Upgrade Guide). The updated policy assured a 2-year transition period for developers to upgrade their apps. As a result, the so-called "breaking" API changes became an important aspect of API governance because these changes either take effect immediately or on short notice (called "90 day breaking changes"). Here, API governance serves to ensure platform stability and predictability in app development for Facebook's growing community of complementors —developers, businesses, marketers, and researchers worldwide—and its growing application ecosystem.

Figure 3 shows the evolution of Facebook's APIs as documented in the changelog going back to v2.1 (2014). The changelog documents the addition of any new features, changes, deprecations (removals),[5] as well as the introduction of App Review requirements, affected nodes (or their fields, edges, parameters, permissions, etc.), and affected API methods (for reading, creating, updating, or deleting Facebook data). As such, it is useful for determining the temporality of API evolution and governance.

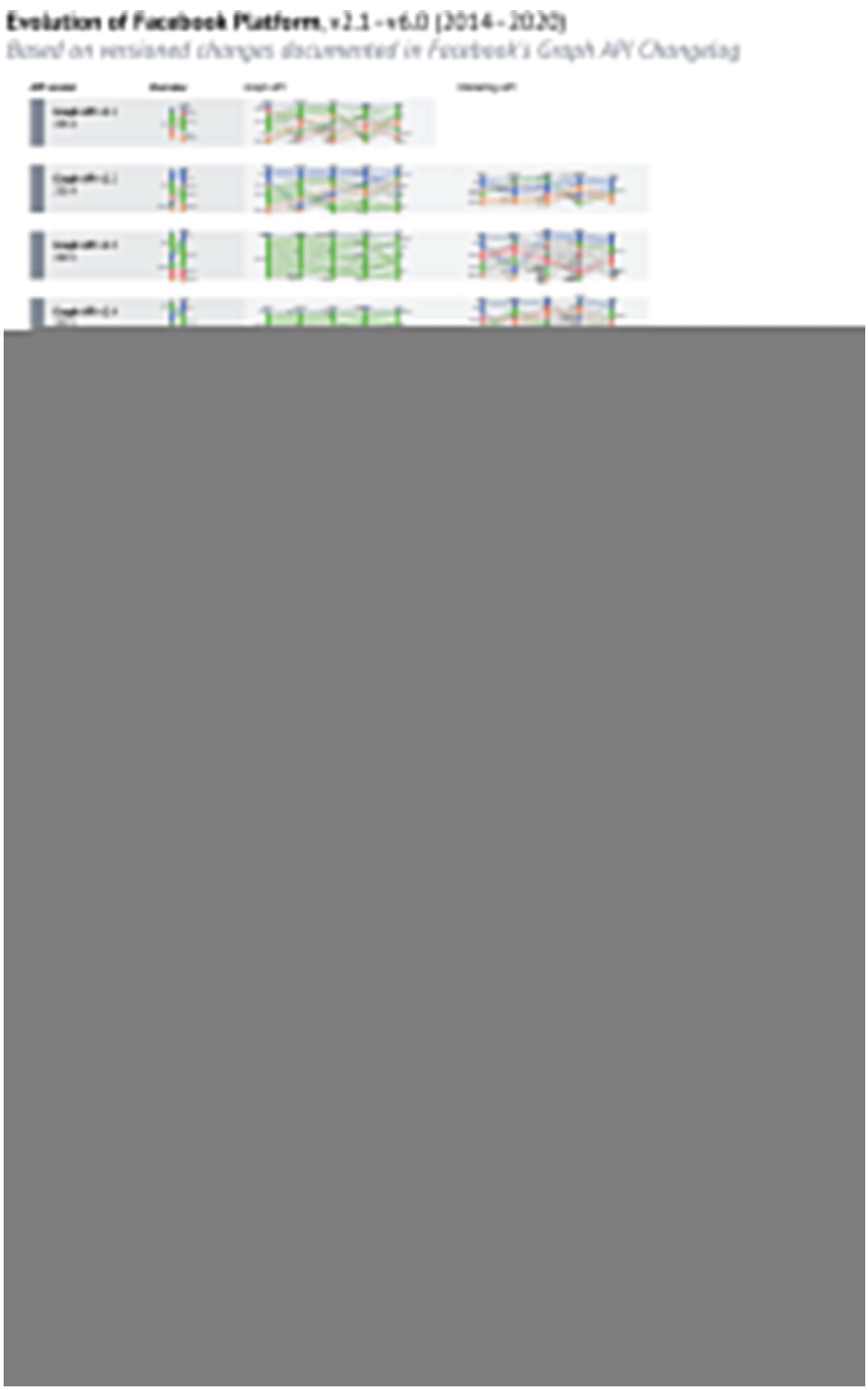


Figure 3. Evolution of Facebook's Graph API Changelog, v2.1–v6.0 (2014–2020) (small multiples). The changelog documents any versioned changes to the Graph and Marketing APIs, and any of Facebook's products or services that rely on them. Flows and color-coding: by change type (e.g., New Features, Changes, Deprecations, etc.) and by API component (e.g., GAPI, MAPI, etc.).

Between 2014–2017 (GAPI v2.1–v2.4; MAPI v2.2–v2.4), Facebook's business side underwent a professionalization and commercialization process. Many additions to the MAPI were made in this period, significantly expanding the data and services accessible to the platform's business developers and marketing partners. Furthermore, new components, such as the Messenger Platform were gradually included in the changelog, reflecting their integration with Facebook's core technical platform.

We observe many changes to the GAPI and MAPI in the wake of public controversies and criticisms of the platform's role regarding the 2016 UK EU-membership referendum, the subsequent US presidential election, and the Fb–CA data scandal (FD-2018c). Facebook introduced additional restrictions on their use, deprecating many available fields, permissions, advertising targeting options, and legacy APIs to improve data protection and permission requirements.[6] In addition, the EU General Data Protection Regulation (GDPR) came into force in May 2018, imposing stricter data protection obligations on Facebook and other firms and organizations that target or collect data related to people in Europe.

The changelog thus also captures how recent stages in Facebook's API evolution were guided by the platform's responses to intensifying pressures from public scrutiny and regulations. So far, these responses have led to changes related to specific API objects but did not alter the internal structure of the API architecture. Many of these changes ended up as "breaking changes" and were announced on a separate (dedicated) page next to the changelog. These breaking changes occur outside of the regular API version release schedule and, as such, they momentarily disrupt the platform's stability. They require the urgent action of app developers who rely on the respective API endpoints. Consequently, breaking changes provide an immediate way for platforms to govern their relationship with developers through their APIs. In 2018–2019, we see that further external pressure demanded additional immediate responses by Facebook and led to many breaking changes. This time, they were related to concerns around discriminatory ad targeting, the Fb–CA scandal, and the GDPR preventing Facebook from using third-party audience data for its self-service advertising tools, which we further discuss in the section on application permissions.

API Object Level: The Graph API User

At the API object level, we see how Facebook Platform defines and represents—that is, datafies—entities as data objects with certain properties ("fields") and connections ("edges"). Object-level API design underpins all of Facebook's apps, including Instagram, WhatsApp, Messenger, and Workplace, because they have all been integrated into the same unified

“data infrastructure” (Nieborg & Helmond, 2019). Object-level design decisions shape how an app can interact with the User object. Moreover, they also impact the platform’s business side because a data object’s fields and edges also serve as targeting options for Facebook’s suite of (both self-service and programmatic) advertising tools, products, and services. As such, Facebook’s API design and governance are entangled with the platform’s data strategy.

Figure 4 presents the evolving composition of the GAPI User object in terms of its fields and edges between 2006 and 2020. Between 2006 and 2010 (before GAPI v1.0), the User is one of seven available API objects, together with the Events, Friends, Messages, Photos, Pokes, and Wall endpoints. The fields of the User object were user-defined inputs that corresponded to the information presented on that user’s profile page (e.g., “about_me,” “gender,” “movies,” “political,” etc.). The number of fields slowly increased during this period, as did the number of edges linked to the User.

Figure 4. Evolution of Facebook’s Graph API User object until v6.0 (2006–2020). The User object represents a user on Facebook as a combination of properties (“fields”) and connections (“edges”). Each vertical slice displays the fields and edges of the User object for one API version.

The release of GAPI v1.0 (2010–2014) marked a turning point for the platform’s data structure because Facebook Platform was subsequently restructured according to the logic of the (social) graph. With this new graph-based data model, data objects, such as the User came to be defined by their connections to other data objects, thus forming relationship networks. The User’s fields (properties) are mostly defined by the users themselves, while its edges (connections) emerge through the user’s online activities, behaviors, and friendships (e.g., /likes for a user’s liked Pages, /friends for a user’s friends). Due to this new data-structuring logic, API changes tend to concern an object’s edges more than its fields. Furthermore, the new graph-based data model also impacted app development and data use (i.e., data access and sharing). It represented Facebook’s vision of a “social” web wherein its users are not only connected to other Facebook data objects but also to data objects outside of the platform’s boundaries. It was during this time that Facebook appealed to third-party app developers to implement its social buttons on their websites and released the Open Graph protocol (2010) to standardize data formats on the web. This was a strategic move that helped make a wealth of external (i.e., unstructured) data sources “platform

ready" to integrate them into Facebook's data infrastructure. The new data model was a pivotal moment in Facebook's evolution from a profile-centric social networking site into an "identity service" (FL-2019d) and a graph-based data infrastructure that could support more than just its social network.

Most changes to the User object between 2014 and 2018 (GAPI v2.0–v2.12) were minor, such as renamed fields and edges. Some of the new edges represent then-launched platform products or features (e.g., /games, /locations, /taggable_friends, and /live_videos). In addition, new fields and edges were introduced for business users and advertisers when the Ads API and GAPI were streamlined in 2014. These new edges thus reflect the User's evolution from a consumer with a profile page to a critical connective node with potentially multiple roles within Facebook's ecosystem (e.g., the user as an app developer, app user, business owner, ad account holder, ad account manager, etc.). In short, the User node became the central gateway through which all of the user's roles are governed by the platform.

In some cases, API objects are governed on the microscopic level of data fields. In early 2018, we first observe that the User's "interested_in" field was no longer available in the MAPI for targeting people in France "due to local laws"(FD-2016) and then in Europe for the same cited reason (FD-2017c) and it was later completely removed (FD-2018a). Similarly, in 2016, *ProPublica* reported that the "ethnic_affinity" field targeting option could be used to create discriminatory housing ads by excluding specific groups, despite this being prohibited according to Facebook's advertising policies (Angwin & Parris, 2016; FP-2021a). Facebook initially updated its policy but later removed the field entirely from the MAPI after ongoing social pressure and multiple lawsuits from civil rights organizations (FD-2017a; FD-2017b; FD-2018h). The issue of discriminatory advertising (or targeting in general) continued for several years, and it was not until 2019, as part of a settlement with civil rights organizations, that housing, employment, and credit ads became a "special ad category" with fewer available targeting options in compliance with US non-discrimination laws (FD-2021k; FNe-2019a; FP-2021b). Nonetheless, a year later, *The Markup* reported that targeting "multicultural affinity categories" was still possible, after which Facebook announced it had removed them in service of "simplifying and streamlining our targeting options" (FB-2020). In short, we witness how the User, as a *targetable data object*, has evolved within the MAPI reference documentation in response to external social and regulatory pressures.

PI v3.0 (April 2018) was the first version to implement major changes on the User object el in the wake of the Fb–CA data scandal disclosed in March 2018. Facebook deprecated

many of the fields associated with the User's profile and restricted the data that apps could access without going through App Review (FD-2018c). However, we see that several fields and edges were not immediately removed or deprecated after the scandal; instead, they would no longer return any data effective immediately (e.g., "about," "education," "interested_in," "political," "relationship_status," "religion," "website," and "work" fields; /friendlists, /taggable_friends, and /mutual_friends edges). Since their immediate removal would break current app distributions that rely on those endpoints, they were not immediately deprecated. Notably, some of these deprecated fields and edges (e.g., demographics, education and workplace, locales, relationship statuses, etc.) remained available as audience-targeting options in Facebook's self-service advertising tools and programmatically through the MAPI (FD-2021j). In other words, while app developers could no longer access certain data through GAPI endpoints, advertisers, marketers, and certified marketing partners could still use them to target users via MAPI endpoints.

Since 2018 (GAPI v3.1–v6.0), there have been no notable changes to the User object except for additional deprecations (e.g., /family, /tagged, /threads, and /notifications edges) "as part of [Facebook's] ongoing commitments to privacy and security" (FD-2020). More importantly, application permissions and the app review process matured as part of Facebook Platform's core governance mechanisms in this period. As we detail next, the User serves a central role in this arrangement.

Application Permissions Level

At the application permissions level, we see how Facebook Platform governs its relationships with complementors (app developers, businesses, academic researchers, etc.) through its APIs. The permissions mediate and structure the relations between platforms and apps, which involve distinct access controls and privileges (e.g., "read-only" to access data, "read/write" to access or modify data, etc.) for different app and user types. The majority of application permissions are now requested through Facebook Login (Figure 5).

Figure 5. Evolution of Facebook's Login Permissions up to v6.0 (2008–2020). Permissions provide a way for applications to access data from Facebook and are mostly requested through Facebook Login. Each vertical slice displays the permissions requested through Facebook Login for one API version.

Permissions for applications did not exist until 2008. Instead, developers had access to "your profile info (excluding contact info), your photos, your events, and most importantly, your friends" by default (FD-2006b). If Facebook users did not want an app—or the apps of their friends—to access their user data, they needed to proactively opt-out in their privacy settings. As such, data access was governed on the consumer side with opt-out privacy settings and not on the developer (or application) side with opt-in permission requests. Extended permissions to access further data were introduced in 2008 for "certain use cases" that "require a greater level of trust from the user" (FD-2008). The extended permissions provided API access for publishing data to the platform on behalf of the app's user (e.g., to send emails, upload photos or videos, or RSVP to Events"). Permissions thus governed the relations between the platform and its connected apps and services, allowing developers to write data to the platform. In 2009, Facebook introduced an optional Application Verification Program (Helmond et al., 2019) to verify an "application's commitment to providing a trustworthy user experience that is secure, respectful and transparent" (FD-2009). Developers were requested to provide basic business information and an explanation of their data requests and data use cases. In return, apps received "verified badges," priority ranking in Facebook's Application Directory, and Facebook advertising credits (FD-2009).

With the release of GAPI v1.0 (2010), Facebook changed the way permissions were granted on the platform, "moving to a model where applications must list all the pieces of data they need to access from a user's profile rather than having all that data available automatically" (FD-2010f). A distinction was introduced between a user's basic (public) profile information (i.e., a person's name, profile picture, gender, username, and friend list), which is visible to all Facebook users and accessible to all apps through the API by default, and a user's private profile information (e.g., "user_likes," "user_religion_politics"), which now required apps to request extended permissions from the user via the new "Granular Permissions Dialog" (FD-2010c; FD-2010e). Consequently, it became more difficult for apps or app developers to access users "sensitive" personal data (a special category under the GDPR).

The platform further restructured its extended permissions into separate /user and /user/friends permissions "to protect the privacy of users who have not explicitly authorized [an] application" (FD-2010b). In the new permissions model, apps could access the basic profile information of a user's friends via the User object without their explicit permission although access to additional friend information required extended permissions. This change also meant that the earlier extended permissions, which initially focused on publishing data to the platform, were expanded with user and friend "data permissions." In

other words, on an individual level, permissions now controlled which apps could read or write data to the platform and which apps could access users' and friends' data. The increasing number of data fields associated with the User object in this period, including new Open GAPI actions, led to a sharp increase in the number of extended permissions—from eight to 49 (2008–2010) to 72 (May 2012). Between 2011 and 2012, there were also new permissions that referred to the Ads API Business User (/business-user) for the first time, reflecting the integration of Facebook's development and business platform governance at the permissions level.

Several notable changes occurred between 2014 and 2018 (GAPI v2.0–v2.12). First, the platform restricted access to users' friends' data in response to mounting concerns over users' data and privacy. The friend list no longer belonged to the set of basic permissions and Facebook now required apps to request extended permission from each app user (FD-2014c). In addition, whereas the GAPI User Friends (/user/friends) endpoint in v1.0 returned lists of users' friends, v2.0 only returned lists of friends who had also installed the app and had given the required permissions. Second, Facebook also launched its current App Review process to ensure that any information obtained by an app is directly connected to a relevant data use case. Moreover, most permission requests now require developers to undergo App Review as well. Facebook informed the UK Competition and Markets Authority (CMA) that this was "aimed at safeguarding users" information against data misuse, leaks, and bad actors (CMA, 2020: Appendix J). Third, a new version of Facebook Login was introduced to handle these application permission requests. The Login Review process was launched to ensure that apps request only those permissions they need (FD-2014a; FD-2021e; FD-2021f). As such, Facebook Login is now used for *authentication* (for users to sign in) as well as for *authorization* (for handling permission requests from apps to access users' information) (FD-2021d). All existing apps needed to comply with the new platform policy or their API access tokens would be revoked.

The new Facebook Login enabled users to make more granular choices about the types or categories of data they wanted to share with third-party apps. However, it also enabled Facebook to evaluate whether the apps submitted for review would add value to the platform's ecosystem. These changes were not only meant to provide more granular privacy controls for Facebook users but were the outcome of internal reassessments at Facebook about the business value of its data-sharing practices with third-party app developers and businesses. This reassessment operation was internally referred to as "protect the graph" (FL-2019e; FL-2019f). As was later revealed, by limiting and restructuring API access to user

and friends data in these ways, Facebook intended to undermine any competitors who used friend data and to reward complementors who added value to Facebook Platform (FL-2019a). Internal documents revealed that App Review was used to determine "the appropriate level of reciprocity" (FL-2019e). Facebook's "reciprocity" principle ("take data, give data") demands that apps "share back to Facebook," so that, it is not only "good for the world," but also "good for us" (FL-2019e). In short, there were now two competing accounts in explaining these API changes: on the one hand, Facebook reports that App Review is a proactive measure for protecting user trust and privacy; on the other hand, it limits competitors' access to valuable Facebook data (e.g., by restructuring the permissions model, revoking API access to friends data, and launching App Review). In Facebook's own words, App Review represented "just another product feature to improve quality," while the API-level changes were meant to "protect the business/model/data" (FL-2019g). As such, the various changes to Facebook's application permissions and the new privacy controls for users in this period served to implement and enforce Facebook's new strategic platform policy (to improve privacy controls and restrict competitors) while monitoring app development. Conversely, as head of Facebook Platform Justin Osofsky wrote: "Historically, we've treated policy enforcement as a secondary function of platform" (FL-2019h).

With the release of GAPI v3.0 (2018), the number of permissions decreased for the first time from 47 to 36. These permissions (e.g., "user_religion_politics," "user_relationships," "read_custom_friendlists," and "user_education_history") were deprecated as a response to the Fb–CA data scandal (FD-2018c; FD-2018f). Furthermore, an increased number of permissions (e.g., "user_friends," "user_likes," and "user_photos") was now restricted to a limited set of partners, not only requiring App Review but also requiring "business verification" and a contract with Facebook (FD-2018d; FD-2018f; FD-2021b; FD-2021c). Facebook Login received its own changelog to document changes to permissions as it had become the core authentication service for both consumers and developers, and a powerful control point for governing app development (FD-2021c). Facebook further increased control over its platform through additional verification processes for individual developers and business entities and required external business-to-business technology providers (i.e., partners) to sign a supplemental terms contract to restrict data usage (FD-2018d; FD-2018e; FD-2018g). In addition, the tiered MAPI access structure for partners and businesses was simplified (FD-2018b). In short, while it may appear as if Facebook Platform only expanded over the years, it also underwent significant restructures at crucial moments in (partial)

response to external social and regulatory pressures from the public and competitive dynamics in the digital platform economy more generally.

Since 2020 (GAPI v8.0–), these permissions have been further streamlined. All permissions were moved onto a separate Permissions Reference page (FD-2021l) and are requested (and thus governed and controlled) through Facebook Login. Permissions other than a user's "public_profile" and email address now require App Review "so that Facebook can confirm that the app uses the data in intended ways and safeguards user privacy" (FD-2021n). This also concerns permissions related to Facebook's other platform instances (e.g., WhatsApp, Messenger, and Instagram apps) and applies to developers, businesses, and creators alike. Only Instagram apps for consumers that require "read-only" access rights to basic profile information, photos, and videos need to request separate Instagram Permissions (FD-2021m). The distinct treatment (i.e., governance) of these app types, we suggest, reflects Facebook's dual governance strategy for business users and creators on one hand and for consumers on the other.

Governance of and by Facebook's APIs

Based on this analysis of Facebook's evolution, we can derive some general features of governance of and by APIs as a contribution to the current literature on platform governance (e.g., Gillespie, 2018; Gorwa, 2019; Gorwa et al., 2020). These features illustrate the ways in which the technicity of platform governance provides control over a platform's ecosystem.

Regarding *governance by* APIs, we must first identify what is governed. Facebook provides several distinct APIs, including those dedicated for (app) development around its core products as well as for advertising and marketing. These distinct interfaces reflect what Facebook's APIs are or are not intended *for* and how those intended uses have changed and evolved. Furthermore, we may distinguish how those things are governed. Facebook Platform has a variety of access controls that serve as additional layers of API governance. These include distinct access levels, rate limits, App Review, verification processes for business and individual developers, and application permissions for distinct app types (e.g., consumer apps, business apps). Furthermore, we may discern who is governed. Facebook offers distinct programming interfaces for its user groups, including the developers of apps, games, and advertising and marketing technology. The distinct user and app types, the App Review process, and permissions provide additional controls over specific (individual) users and uses. Finally, we may determine when governance or control is exercised by APIs. Most

changes to Facebook's APIs are now versioned and documented, including any breaking changes, which can go into effect either immediately or after 90 days and later, giving developers some time to update their app infrastructure or accept that their app (or business) is no longer viable. The developer documentation and blog posts typically contextualize these changes, explaining why they are happening according to Facebook.

Regarding *governance of* APIs, we observe that API changes can be the outcome of both internal (e.g., policy changes and strategic decisions) or external pressures (e.g., social, competitive, or regulatory). The aforementioned significant events were followed by many changes to Facebook's APIs, especially on the object and application permissions levels. Similarly, ongoing social and regulatory pressures from civil rights organizations and journalists regarding discriminatory ad targeting and the introduction of new regulatory frameworks, such as the GDPR led to observable API changes. Finally, leaked files from app development firm Six4Three's lawsuit against Facebook revealed how the social platform allegedly implemented API changes and App Review to "protect" its business model while strategically controlling and closing down competitors (FL-2019g).

These features of governance of and by platforms through APIs are by no means comprehensive. However, they illustrate how APIs are not merely technical objects used by developers to access or retrieve a platform's data and services. Because of their strategic importance, APIs like those of Facebook, have evolved into complex layered and interconnected governance arrangements that shape the conditions for app and business development in often subtle ways through coordination, managerial, and control mechanisms (e.g., Eaton et al., 2015; Stratton, 2020). These mechanisms govern the platform's many different types of users and uses, including business and academic research. In this way, Facebook's API governance shares similarities with what Caplan and Gillespie (2020) describe as a "tiered governance" strategy in the case of YouTube, where different users face different rules, material resources, and procedures (p. 6).

In the following, we discuss the implications of our analysis concerning API governance and platforms' infrastructural power. Our observation that APIs have evolved into complex governance arrangements and function as more than technical objects for data access or retrieval prompt a discussion into the relationship between the technicity of platform governance—governance of and by APIs—and the material conditions of a platform's power.

API Evolution, Governance, and Infrastructural Power

Significantly, API governance targets the conduct of a platform's complements and complementors in the platform's ecosystem more so than the conduct of its consumers. As such, the analysis augments the literature on platform governance, which often focuses on governance over consumers (e.g., content moderation, algorithms, policies, etc.) or content producers, and may or may not involve an interest in a platform's technicity (e.g., Caplan & Gillespie, 2020; Gillespie, 2018; Gorwa et al., 2020; Medzini, 2021). However, APIs are a critical part of platform governance because they constitute the material conditions of platforms and apps (cf. Blanke & Pybus, 2020; Gerlitz et al., 2019; van der Vlist & Helmond, 2021). They allow third parties to develop apps and services "on top" of a platform while enabling the platform to maintain infrastructural control over those apps and services, thus concurrently enlarging a platform's operational scale and scope while consolidating its position of power within the ecosystem.

This analysis traced how Facebook's APIs have evolved from a relatively simple technical object (programming interface) into a complex arrangement of technical objects, specifications, terms and policies, and review and verification processes to govern the platform's diverse users and uses in different ways. Application permissions do not only enable distinct and approved data transactions between a platform and its complementors (Pybus & Coté, 2021) but also govern those transactions and complementor relationships in increasingly granular ways. In addition, increasingly granular governance mechanisms, including App Review and verification processes for individuals and businesses, may improve privacy or security but are also used to block "bad actors" and unwanted competitors. In short, APIs are far more than mere developer tools; the more granular this governance arrangement becomes, the more we need to caution against potential (ab)uses of a platform's power in the ecosystem.

If platforms like Facebook exert a powerful influence over their ecosystem, then we need to consider the implications for those affected. We observe that platforms like Facebook do not only shape their own evolution (e.g., Helmond et al., 2019), but also shape (or "orchestrate") the evolution of their ecosystem through API design, governance, and strategy. It is because APIs are governance arrangements, more than mere programming interfaces, that platforms can set, shape, and alter the material conditions of development on their different "sides," such as for app development by individuals, marketers, or advertisers. These conditions directly influence, often in subtle ways, what can and cannot be built, sustained, or thrive in [illegible] ecosystem. Platforms' boundary resources play a central role in this because they [illegible]blish the material conditions of participation and control. On an infrastructural level, API

design shapes what is technically feasible on a given platform, while a platform's governance shapes what is allowed, encouraged, or technically and economically viable for those within that ecosystem. With Facebook, we note that certain uses of its data and services were allowed and encouraged (e.g., building "rich social apps"), whereas other uses were eventually discouraged or restricted (e.g., "data export tools").

Platforms also control how, when, and for whom these conditions change, which they can enact as they see fit, often without risking their position of power. Although Facebook Platform may allow diverse user and stakeholder groups to participate in its ecosystem, it also ensures that those complementors are not equal in their ability to influence the platform owner or other complementors, resulting in asymmetries and different degrees of agency (Eaton et al., 2015, p. 219). Any change may cause disturbances or ripple effects across the entire ecosystem of apps and services relying on an API, potentially impacting the viability of all apps and services supported or sustained by it, including those of businesses and academic researchers. When Facebook changed API access to friends' data in 2014–2015, this severely impacted the business models and apps of complementors as well as academic research tools, causing shutdowns across the ecosystem (e.g., Constine, 2015; FL-2019b).

This dimension of a platform's power, which puts special emphasis on the material or infrastructural conditions of development, may be conceptualized as an evolutive aspect of the "infrastructural power" of digital platforms (cf. Busch et al., 2021). We thus suggest that the analysis of platforms' power may include a specific focus on their unique ability to influence or "orchestrate" not only their own evolution but also the evolution of the larger ecosystem (Eaton et al., 2015; Stratton, 2020; Tiwana, 2014). In addition, while one may study how platforms or APIs change and evolve, we suggest considering how powerful platforms like Facebook wield API changes as a tool for governance and control. Evolution concerns not only inevitable changes but also strategy.

This strategic dimension is perhaps most clearly visible on the API architecture level: Facebook's APIs went through multiple cycles of diversification and integration with the continuous addition of new endpoints and the consolidation of new API components into its core technical platform, respectively. While some of these new components and endpoints originated from Facebook's own internal development (e.g., Messenger), others notably originated either from mergers and acquisitions (e.g., Instagram, WhatsApp, Atlas Solutions, Rail, etc.) or from Facebook's marketing partnership strategy (e.g., the MAPIs) (Helmond et al., 2019). These different components were all eventually integrated into Facebook's core

technical platform, although the process took several years. Integrations initially occurred in the "back-end," enabling developers and businesses to reap the benefits of a consolidated, unified data infrastructure. Meanwhile, Facebook to this day has maintained a fragmented front-end for consumers: it offers a "family of apps" wherein each one speaks to a different segment of the platform's user population or accommodates a different set of user practices.[7] The consolidation of these different apps and services "from Facebook," including their different APIs, may also make it increasingly difficult to break up the platform. In addition, the diversification and integration processes tie into "platform capture" (Partin, 2020). Facebook has leveraged these asymmetries in its complementor relationships, demanding that complementors not only "take data" but also "give data" back to Facebook when using its APIs. Finally, this diversification process has enabled large platforms like Facebook and Google to decompose and recompose themselves into what Blanke and Pybus describe as "service assemblages." This process has led to "a much deeper technical integration" of application ecosystems, enabling Facebook and Google "to shift the dynamics of competition and monopolization in their favour" (Blanke & Pybus, 2020, p. 2). In sum, the ongoing processes of diversification and integration provide important starting points for further analyses of platforms' evolutive and infrastructural power.

Conclusion

This article has traced the evolution of Facebook's APIs, which evolved from a relatively simple programming interface for data access into a complex layered and interconnected governance arrangement that links API design, governance, and strategy. Since its launch in 2006, Facebook Platform has provided many different APIs that both facilitate and govern the material conditions of app development and the social and economic processes they sustain. As observed, APIs have played many different roles, not only as developer tools but also as a means of enforcing platform policy and strategy and influencing the evolution of the wider ecosystem in often subtle ways. Studying the coevolution of API design and governance thus provides important insights into determining how platforms secure and maintain infrastructure control and how this is operationalized or evolving over time. As such, this study is an empirical-historical contribution to begin understanding the infrastructural power "blindspot" in the academic and policy debates on platform power (Busch et al., 2021).

Previous reports stressed the need for better understanding the infrastructural aspects of platform power, and for further research on this topic. We have argued that the material

conditions and evolution of APIs can be analyzed to develop such a better understanding of the infrastructural aspects of platform power. Specifically, we have demonstrated that APIs are not a single or monolithic source of power, but a complex arrangement of many different governance and control mechanisms that, together, represent a key source of infrastructural power. These mechanisms target many different uses and user groups, including third-party app developers, businesses, and partners. They also provide centralized and unidirectional hierarchical control over large numbers of apps and services—and the developers who build and maintain them— built "on top" of the platform's APIs. Moreover, they represent a source of power over the evolution of the platform's ecosystem as a whole, which, in the case of Facebook, comprises millions of complements and complementors.

The multiple levels of analysis that we distinguished illustrate the complexity of APIs as technical objects with respect to the technicity of platform governance. Many of the different features and mechanisms of governance and coordination that we found across these levels are subtle: they are "powerful precisely because it is not a grand and spectacular strategy but a functional and often invisible reality" (Munn, 2020, p. 15). They shape the material conditions and evolution of apps and services connected to Facebook Platform in specific ways. Given this significance, it is important to continue to consider how large platforms like Facebook evolve and how they negotiate external pressures to change and reform on the level of the technical platform. Independently archived developer pages and API reference documentation provide necessary empirical materials for this purpose, and can help address the current lack of observability regarding complex technical systems and digital platforms (Rieder & Hofmann, 2020). This may also help policymakers and competition and regulation authorities by providing insights into Facebook's data-sharing practices with different types of third parties.

Furthermore, our approach enables studying what Facebook, as one of the most popular social media platforms, is built and intended for, and how this has changed and evolved over the years. Specifically, we have demonstrated the value and utility of archived web sources for reconstructing exactly how Facebook has decomposed and recomposed itself as a platform for developers, enabling a comprehensive view of the platform as a "service assemblage" (cf. Blanke & Pybus, 2020; Helmond et al., 2019). Moreover, such a comprehensive view is difficult, if not impossible, to obtain otherwise due to the many serious issues and challenges faced by social media and platform historians and archivists (cf., Helmond & van der Vlist, 2019).

The large corpus of developer pages and API reference documentation used in this study may guide and inspire further research into the history, evolution, and importance of APIs in relation to a platform's governance and infrastructural power. Specifically, we recommend further research on APIs and (the technicity of) platform governance to further explore the material conditions and evolutionary dynamics of powerful platforms like Facebook who occupy a unique position of power within the ecosystem. We encourage scholars to use similar approaches to study the evolution and material conditions of platform governance and how APIs tie into platforms' infrastructural power. For example, comparative studies of APIs and their evolution may provide relevant insights into the distinctive governance arrangements they represent. Further research could look into how the API mechanisms that we identified compare across different platforms and how they figure into broader theoretical discussions about infrastructural power.

We also need more comprehensive views of the application ecosystems that are connected to platforms' APIs (and thus impacted by their exercise of governance and infrastructural power). The most popular APIs are particularly prone to cause large-scale ripple effects and potential infrastructure breakdowns, which may extend throughout the entire ecosystem of apps and services and the various social and economic processes they support or sustain. Finally, we stress that web archives play a vital role in preserving the material traces necessary to reconstruct a platform's evolution or history, despite the laborious challenge of studying platforms and APIs that constantly change.

Acknowledgments

The authors thanks Ángeles Briones (DensityDesign Lab, Polytechnic University of Milan) for information visualization assistance during the research process, and the Center for Advanced internet Studies (CAIS) for funding and hosting our working group. They also thank José van Dijck, Tobias Blanke, Thomas Poell, and the anonymous reviewers, whose constructive feedback improved the article.

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: This work was supported by the German Research Foundation (Deutsche *Forschungsgemeinschaft*, DFG), project number 262513311 (SFB 1187: “Media of Cooperation”); and the research program Innovational Research Incentives Scheme Veni with project number 275-45-009, which is (partly) financed by the Dutch Research Council (NWO). We further received funding from the Center for Advanced internet Studies (CAIS), working group “Data sharing troubles: Tracing the evolution of the Facebook Graph API” (2019–2020).

ORCID iDs

Fernando N. van der Vlist

Anne Helmond

Footnotes

1. Facebook has launched several (controversial) API-based initiatives since 2018 to build academic partnerships and improve transparency, including SOCIAL SCIENCE ONE, the Facebook Ad Library (formerly Ad Archive), and the FORT Pages API.

Go To Footnote

2. Facebook Platform is “the set of APIs, SDKs, tools, plugins, code, technology, content, and services that enables others, including app developers and website operators, to develop functionality, retrieve data from Facebook and any other Facebook Products, or provide data to us” (FD-2021o).

Go To Footnote

3. Available at: https://web.archive.org/web/*/developers.facebook.com/* (2006–2020) and https://web.archive.org/web/*/wiki.developers.facebook.com/* (2007–2011). This count only includes those “snapshots” with a HTTP 200 OK success status response code, which indicates that the request has succeeded.

Go To Footnote

4. REST (representational state transfer) is a software architectural style that uses HTTP-based methods for requests and responses (e.g., “GET,” “POST,” “DELETE,” etc.). It is most commonly used to create interactive apps on the web.

Go To Footnote

5. “Deprecation” refers to the (scheduled) removal of nodes or edges, even if replacements were introduced simultaneously. In the latter case, we use the label “Replacement” instead.

Go To Footnote

6. While we found some examples of this, it is not always the case that API changes due to regulations, such as the GDPR are explicitly motivated in the reference documentation. Instead, their context is typically provided in separate accompanying posts on Facebook’s Developer Blog.

Go To Footnote

7. As a result, many users are unaware that Facebook also owns Instagram, WhatsApp, and other popular apps. From November 2019, the company’s rebranded “from FACEBOOK” tagline makes Facebook’s ownership of all these apps much more evident (Constine, 2019; FNe-2019b).

Go To Footnote

Data availability statement

The data that support the findings of this study are openly available in the Open Science Framework (OSF) at https://doi.org/10.17605/osf.io/wfxyp. Additionally, PyCatFlow (a Python package for visualizing categorical time series data used to create Figures 4 and 5) is available at https://doi.org/10.5281/zenodo.5531785.

Appendix

Table 1. References to Original and Archived Web Sources.

In-text citation	Type	Reference
FB-2020	Business	(2020, August 11). Simplifying targeting categories. https://www.facebook.com/business/news/update-to-facebook-ads-targeting-categories/
FB-2021*	Business	(n.d.). Facebook ads: Online advertising on Facebook. https://www.facebook.com/business/ads
FD-2006a	Developers	(2006, October 10). Facebook Developers News. https://web.archive.org/web/20061009080928/https://developers.facebook.com/news.php
FD-2006b	Developers	(2006, December 13). A user's overview of the Facebook Development Platform. https://web.archive.org/web/20061213220159/http://developers.facebook.com/background.php
FD-2007	Developers	(2007, February 17). Extended permissions. https://web.archive.org/web/20070217011936/http://developers.facebook.com/news.php
FD-2008	Developers	(2008, September 6). Extended permissions. https://web.archive.org/web/20080906150906/http://wiki.developers.facebook.com/index.php/Extended_permissions
FD-2009	Developers	(2009, June 18). Application Verification Program. https://web.archive.org/web/20090618153142/http://developers.facebook.com/verification.php?tab = faq
FD-2010a	Developers	Purdy, D. (2010a, October 16). Operation Developer Love. https://developers.facebook.com/blog/post/417/
-2010b	Developers	(2010b, April 23). Extended permissions. https://web.archive.org/web/20100423170819/http://developers.faceb

In-text citation	Type	Reference
		ook.com/docs/authentication/permissions
FD-2010c	Developers	(2010c, May 8). Extended permissions. https://web.archive.org/web/20100508225433/developers.facebook.com/docs/authentication/permissions
FD-2010d	Developers	(2010d, August 31). Developer roadmap. https://web.archive.org/web/20100831180503/http://developers.facebook.com/roadmap
FD-2010e	Developers	(2010e, December 5). Authentication. https://web.archive.org/web/20101205122001/http://developers.facebook.com/docs/authentication/
FD-2010f	Developers	(2010f, December 12). Platform upgrade guide. https://web.archive.org/web/20101205155259/http://developers.facebook.com/docs/guides/upgrade
FD-2011a	Developers	(2011a, August 31). A stable platform. https://developers.facebook.com/blog/post/550/
FD-2011b	Developers	(2011b, October 11). Breaking change policy. https://web.archive.org/web/20111011011437/https://developers.facebook.com/roadmap/change-policy/
FD-2014a	Developers	Spehar, J. (2014a, April 30). The new Facebook Login and Graph API 2.0. https://developers.facebook.com/blog/post/2014/04/30/the-new-facebook-login/
FD-2014b	Developers	Sharma, M. (2014b, October 30). Graph API v2.2 and updated iOS and Android SDKs. https://developers.facebook.com/blog/post/2014/10/30/graph-api-v2.2/
FD-2014c	Developers	(2014c, May 21). Facebook platform changelog. https://web.archive.org/web/20140521164802/https://developers.facebook.com/docs/apps/changelog
-2016	Developers	(2016, January 20). Targeting specs. https://web.archive.org/web/20160120033705if_/https://developers.fac

In-text citation	Type	Reference
		ebook.com/docs/marketing-api/targeting-specs/v2.4
FD-2017a	Developers	Oakes Dunn, A. (2017a, September 20). Self-reported targeting. https://developers.facebook.com/ads/blog/post/2017/09/20/self-reporting-targeting/
FD-2017b	Developers	Bala, D. (2017b, December 4). Targeting exclusions. https://developers.facebook.com/ads/blog/post/v2/2017/12/04/targeting-exclusions-blog-post/
FD-2017c	Developers	(2017c, February 25). Targeting specs. https://web.archive.org/web/20170225150830/https://developers.facebook.com/docs/marketing-api/targeting-specs
FD-2018a	Developers	Oakes Dunn, A. (2018a, February 28). Changes to targeting availability for “interested in.” https://developers.facebook.com/ads/blog/post/2018/02/27/targeting-availability-interestedin/
FD-2018b	Developers	Chen, J. (2018b, July 2). Marketing API tier simplification. https://developers.facebook.com/ads/blog/post/v2/2018/07/02/marketing-api-tier-simplification/
FD-2018c	Developers	Archibong, I. (2018c, April 4). API and other Platform product changes. https://developers.facebook.com/blog/post/2018/04/04/facebook-api-platform-product-changes/
FD-2018d	Developers	Papamiltiadis, K. (2018d, May 1). Enhanced developer App Review and Graph API 3.0 now live. https://developers.facebook.com/blog/post/2018/05/01/enhanced-developer-app-review-and-graph-api-3.0-now-live/
FD-2018e	Developers	Papamiltiadis, K. (2018e, December 10). Facebook launches verification for individual developers. https://developers.facebook.com/blog/post/2018/12/10/verification-for-individual-developers/
-2018f	Developers	(2018f, April 4). https://developers.facebook.com/docs/graph-api/changelog/non-versioned-changes/apr-4-2018#login-4-4

In-text citation	Type	Reference
FD-2018g	Developers	(2018g, May 1). Version 3.0: App Review. https://developers.facebook.com/docs/graph-api/changelog/version3.0#new-app-review
FD-2018h	Developers	(2018h, March 9). Targeting. https://web.archive.org/web/20180309021752/https://developers.facebook.com/docs/marketing-api/buying-api/targeting
FD-2020	Developers	(2020, February 3). Version 6.0. https://developers.facebook.com/docs/graph-api/changelog/version6.0
FD-2021a*	Developers	(n.d.). Developer policies. https://developers.facebook.com/devpolicy/
FD-2021b*	Developers	(n.d.). Release. https://developers.facebook.com/docs/development/release
FD-2021c*	Developers	(n.d.). Changelog. https://developers.facebook.com/docs/facebook-login/changelog
FD-2021d*	Developers	(n.d.). Overview. https://developers.facebook.com/docs/facebook-login/overview
FD-2021e*	Developers	(n.d.). Permissions with Facebook Login. https://developers.facebook.com/docs/facebook-login/permissions/overview
FD-2021f*	Developers	(n.d.). App Review. https://developers.facebook.com/docs/facebook-login/review
FD-2021g*	Developers	(n.d.). Changelog. https://developers.facebook.com/docs/graph-api/changelog/
FD-2021h*	Developers	(n.d.). Overview. https://developers.facebook.com/docs/graph-api/overview/
-2021i*	Developers	(n.d.). Graph API user. https://developers.facebook.com/docs/graph-api/reference/user/

In-text citation	Type	Reference
FD-2021j*	Developers	(n.d.). Audiences. https://developers.facebook.com/docs/marketing-api/audiences
FD-2021k*	Developers	(n.d.). Special ad category. https://developers.facebook.com/docs/marketing-api/audiences/special-ad-category/
FD-2021l*	Developers	(n.d.). Permissions reference. https://developers.facebook.com/docs/permissions/reference
FD-2021m*	Developers	(n.d.). Permissions reference: Instagram permissions. https://developers.facebook.com/docs/permissions/reference#instagram_permissions
FD-2021n*	Developers	(n.d.). Permissions reference: Facebook login permissions. https://developers.facebook.com/docs/permissions/reference#login_permissions
FD-2021o*	Developers	(n.d.). Facebook platform terms. https://developers.facebook.com/terms/
FL-2019a	Leaks	Campbell, D. (2019, November 6). Facebook leaks. https://www.duncancampbell.org/facebookleaks
FL-2019b	Leaks	Campbell, D. (2019, November 6). Facebook leaks (pp. 2–70). https://dataviz.nbcnews.com/projects/20191104-facebook-leaked-documents/assets/facebook-sealed-exhibits.pdf
FL-2019c	Leaks	Campbell, D. (2019, November 6). Facebook leaks: Exhibit 43 (p. 1064). https://dataviz.nbcnews.com/projects/20191104-facebook-leaked-documents/assets/facebook-sealed-exhibits.pdf
FL-2019d	Leaks	Campbell, D. (2019, November 6). Facebook leaks: Exhibit 62 (p. 962). https://dataviz.nbcnews.com/projects/20191104-facebook-leaked-documents/assets/facebook-sealed-exhibits.pdf
-2019e	Leaks	Campbell, D. (2019, November 6). Facebook leaks: Exhibit 78 (p. 802). https://dataviz.nbcnews.com/projects/20191104-facebook-leaked-

In-text citation	Type	Reference
		documents/assets/facebook-sealed-exhibits.pdf
FL-2019f	Leaks	Campbell, D. (2019, November 6). Facebook leaks: Exhibit 104 (pp. 1356–1393). https://dataviz.nbcnews.com/projects/20191104-facebook-leaked-documents/assets/facebook-sealed-exhibits.pdf
FL-2019g	Leaks	Campbell, D. (2019, November 6). Facebook leaks: Exhibit 125 (p. 1460). https://dataviz.nbcnews.com/projects/20191104-facebook-leaked-documents/assets/facebook-sealed-exhibits.pdf
FL-2019h	Leaks	Campbell, D. (2019, November 6). Facebook leaks: Exhibit 175 (p. 3482). https://dataviz.nbcnews.com/projects/20191104-facebook-leaked-documents/assets/facebook-sealed-exhibits.pdf
FNe-2019a	Newsroom	Sandberg, S. (2019a, March 19). Doing more to protect against discrimination in housing, employment and credit advertising. https://about.fb.com/news/2019/03/protecting-against-discrimination-in-ads/
FNe-2019b	Newsroom	Lucio, A. (2019b, November 4). Introducing our new company brand. https://about.fb.com/news/2019/11/introducing-our-new-company-brand/
FP-2021a*	Policies	(n.d.). Advertising Policies. https://www.facebook.com/policies/ads/
FP-2021b*	Policies	(n.d.). Advertising Policies: 3. Discriminatory practices. https://www.facebook.com/policies/ads/prohibited_content/discriminatory_practices
IE-2014	Engineering	(2014, June 25). Migrating from AWS to FB. https://instagram-engineering.com/migrating-from-aws-to-fb-86b16f6766e2

* Date accessed 23 March 2021.

References

Alaimo C., Kallinikos J. (2019). Social media and the infrastructuring of sociality. In Kornberger M., Bowker G. C., Elyachar J., Mennicken A., Miller P., Randa Nucho J., Pollock N. (Eds.), *Thinking Infrastructures* (pp. 289–306). Emerald Publishing. https://doi.org/10.1108/S0733-558X20190000062018

Go To Reference

Google Scholar

Alaimo C., Kallinikos J., Valderrama E. (2020). Platforms as service ecosystems: Lessons from social media. *Journal of Information Technology*, 35(1), 25–48. https://doi.org/10.1177/0268396219881462

+ Show References

Google Scholar

Albright J. (2018, March 21). The graph API: Key points in the Facebook and Cambridge Analytica debacle. *Medium*. https://medium.com/tow-center/the-graph-api-key-points-in-the-facebook-and-cambridge-analytica-debacle-b69fe692d747

+ Show References

Google Scholar

Angwin J., Parris T. (2016, October 28). Facebook lets advertisers exclude users by race. *ProPublica*. https://www.propublica.org/article/facebook-lets-advertisers-exclude-users-by-race

Go To Reference

Google Scholar

Barrett B., Kreiss D. (2019). Platform transience: Changes in Facebook's policies, procedures, and affordances in global electoral politics. *Internet Policy Review*, 8(4). https://doi.org/10.14763/2019.4.1446

Go To Reference

Google Scholar

Bechmann A. (2013). Internet profiling: The economy of data intraoperability on Facebook and Google. *MedieKultur*, 29(55), 72–91.

Go To Reference

Crossref

Google Scholar

Blanke T., Pybus J. (2020). The material conditions of platforms: Monopolization through decentralization. *Social Media + Society*, 6(4), 1–13. https://doi.org/10.1177/2056305120971632

+ Show References

Google Scholar

Bodle R. (2011). Regimes of sharing: Open APIs, interoperability, and Facebook. *Information, Communication & Society*, 14(3), 320–337. https://doi.org/10.1080/1369118X.2010.542825

Go To Reference

Google Scholar

Bruns A. (2019). After the 'APIcalypse': Social media platforms and their fight against critical scholarly research. *Information, Communication & Society*, 22(11), 1544–1566. https://doi.org/10.1080/1369118X.2019.1637447

Go To Reference

Google Scholar

Bucher T. (2013). Objects of intense feeling: The case of the Twitter API. *Computational Culture*, 3. http://computationalculture.net/article/objects-of-intense-feeling-the-case-of-the-t

witter-api

+ Show References

Google Scholar

Busch C. (2021). *Regulation of digital platforms as infrastructures for services of general interest*. Friedrich-Ebert-Stiftung. https://library.fes.de/pdf-files/wiso/17836.pdf

Go To Reference

Google Scholar

Busch C., Graef I., Hofmann J., Gawer A. (2021). *Uncovering blindspots in the policy debate on platform power*. Expert Group for the Observatory of the Online Platform Economy. https://platformobservatory.eu/app/uploads/2021/03/05Platformpower.pdf

+ Show References

Google Scholar

Caplan R., Gillespie T. (2020). Tiered governance and demonetization: The shifting terms of labor and compensation in the platform economy. *Social Media + Society*, 6(2), 1–13. https://doi.org/10.1177/2056305120936636

+ Show References

Google Scholar

Competition and Markets Authority (CMA). (2020). *Online platforms and digital advertising market study*. https://www.gov.uk/cma-cases/online-platforms-and-digital-advertising-market-study

+ Show References

Google Scholar

Constine J. (2015, April 28). Facebook is shutting down its API for giving your friends' data to apps. *TechCrunch*. https://social.techcrunch.com/2015/04/28/facebook-api-shut-down/

Go To Reference

Google Scholar

Constine J. (2019, November 4). Facebook's new branding distinguishes app from acquisitions. *TechCrunch*. https://social.techcrunch.com/2019/11/04/facebook-branding/

Go To Reference

Google Scholar

Dal Bianco V., Myllärniemi V., Komssi M., Raatikainen M. (2014). The role of platform boundary resources in software ecosystems: A case study. In *WICSA'14* (pp. 11–20). New York: IEEE. https://doi.org/10.1109/WICSA.2014.41

Go To Reference

Google Scholar

DeNardis L., Hackl A. M. (2015). Internet governance by social media platforms. *Telecommunications Policy*, 39(9), 761–770. https://doi.org/10.1016/j.telpol.2015.04.003

Go To Reference

Google Scholar

de Souza C. R. B., Redmiles D., Cheng L.-T., Millen D., Patterson J. (2004). How a good software practice thwarts collaboration: The multiple roles of APIs in software development. In *SIGSOFT'04/FSE'12* (pp. 221–230). New York: ACM. https://doi.org/10.1145/1029894.1029925

Go To Reference

gle Scholar

Duguay S., Burgess J., Suzor N. (2020). Queer women's experiences of patchwork platform governance on Tinder, Instagram, and Vine. *Convergence*, 26(2), 237–252. https://doi.org/10.1177/1354856518781530

Go To Reference

Google Scholar

Eaton B., Elaluf-Calderwood S., Sørensen C., Yoo Y. (2015). Distributed tuning of boundary resources: The case of Apple's iOS service system. *MIS Quarterly*, 39(1), 217–243. https://doi.org/10.25300/MISQ/2015/39.1.10

+ Show References

Google Scholar

Evans P. C., Basole R. C. (2016). Revealing the API ecosystem and enterprise strategy via visual analytics. *Communications of the ACM*, 59(2), 26–28. https://doi.org/10.1145/2856447

+ Show References

Google Scholar

Federal Trade Commission. (2019a, July 24). *FTC's $5 billion Facebook settlement: Record-breaking and history-making*. https://www.ftc.gov/news-events/blogs/business-blog/2019/07/ftcs-5-billion-facebook-settlement-record-breaking-history

Go To Reference

Google Scholar

Federal Trade Commission. (2019b, December 9). *FTC issues opinion and order against Cambridge Analytica for deceiving consumers about the collection of Facebook data, compliance with EU-U.S. privacy shield*. https://www.ftc.gov/news-events/press-releases/2019/12/ftc-issues-opinion-order-against-cambridge-analytica-deceiving

Go To Reference

Google Scholar

Federal Trade Commission. (2020, December 9). *FTC sues Facebook for illegal monopolization*. https://www.ftc.gov/news-events/press-releases/2020/12/ftc-sues-facebook-illegal-monopolization

Go To Reference

Google Scholar

FT Reporters. (2020, July 28). Big Tech goes to Washington. *Financial Times*. https://www.ft.com/content/3e26d31f-9cff-4b3b-a971-02e16996c190

Go To Reference

Google Scholar

Gerlitz C., Helmond A. (2013). The Like economy: Social buttons and the data-intensive web. *New Media & Society*, 15(8), 1348–1365. https://doi.org/10.1177/1461444812472322

+ Show References

Google Scholar

Gerlitz C., Helmond A., van der Vlist F. N., Weltevrede E. (2019). Regramming the platform: Infrastructural relations between apps and social media. *Computational Culture*, 7. http://computationalculture.net/regramming-the-platform/

+ Show References

Google Scholar

Ghazawneh A., Henfridsson O. (2012). Balancing platform control and external contribution in third-party development: The boundary resources model. *Information Systems Journal*, 23(2), 173–192. https://doi.org/10.1111/j.1365-2575.2012.00406.x

Go To Reference

Google Scholar

Gillespie T. (2018). Regulation of and by platforms. In Burgess J., Poell T., Marwick A. (Eds.), *The SAGE handbook of social media* (pp. 196–212). SAGE.

+ Show References

Crossref

Google Scholar

Gorwa R. (2019). What is platform governance? *Information, Communication & Society*, 22(6), 854–871. https://doi.org/10.1080/1369118X.2019.1573914

+ Show References

Google Scholar

Gorwa R., Binns R., Katzenbach C. (2020). Algorithmic content moderation: Technical and political challenges in the automation of platform governance. *Big Data & Society*, 7(1), 1–15. https://doi.org/10.1177/2053951719897945

+ Show References

Google Scholar

Greene D., Shilton K. (2017). Platform privacies: Governance, collaboration, and the different meanings of “privacy” in iOS and Android development. *New Media & Society*, 20(4), 1640–1657. https://doi.org/10.1177/1461444817702397

Go To Reference

Google Scholar

Helmond A. (2015). The platformization of the web: Making web data platform ready. *Social Media + Society*, 1(2), 1–11. https://doi.org/10.1177/2056305115603080

+ Show References

Google Scholar

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. https://doi.org/10.1080/24701475.2019.1593667

+ Show References

Google Scholar

Helmond A., van der Vlist F. N. (2019). Social media and platform historiography: Challenges and opportunities. *TMG—Journal for Media History*, 22(1), 6–34. https://doi.org/10.18146/tmg.434/

+ Show References

Google Scholar

Ho J. C.-T. (2020). How biased is the sample? Reverse engineering the ranking algorithm of Facebook's Graph application programming interface. *Big Data & Society*, 7(1), 1–15. https://doi.org/10.1177/2053951720905874

Go To Reference

Google Scholar

Hogan B. (2018). Social media giveth, social media taketh away: Facebook, friendships, and APIs. *International Journal of Communication*, 12, Article 20.

Go To Reference

Google Scholar

[illegible] M. (2019, January 25). Zuckerberg plans to integrate WhatsApp, Instagram and Facebook Messenger. *The New York Times*. https://www.nytimes.com/2019/01/25/technology/

facebook-instagram-whatsapp-messenger.html

Go To Reference

Google Scholar

Iyer B., Getchell K. (2018, February 13). Why APIs should be regulated. *MIT Sloan Management Review*. https://sloanreview.mit.edu/article/why-regulate-digital-organizations-apis/

+ Show References

Google Scholar

John N. A., Nissenbaum A. (2018). An agnotological analysis of APIs: Or, disconnectivity and the ideological limits of our knowledge of social media. *The Information Society*, 35(1), 1–12. https://doi.org/10.1080/01972243.2018.1542647

Go To Reference

Google Scholar

Lahey M. (2016). Invisible actors Web application programming interfaces, television, and social media. *Convergence*, 22, 426–439. https://doi.org/10.1177/1354856516641915

Go To Reference

Google Scholar

Langlois G., Elmer G. (2019). Impersonal subjectivation from platforms to infrastructures. *Media, Culture & Society*, 41(2), 236–251. https://doi.org/10.1177/0163443718818374

Go To Reference

Google Scholar

y S. (2020). *Facebook: The inside story*. Blue Rider Press.

+ Show References

Google Scholar

Medjaoui M., Wilde E., Mitra R., Amundsen M. (2018). *Continuous API management: Making the right decisions in an evolving landscape*. O'Reilly.

+ Show References

Google Scholar

Medzini R. (2021). Enhanced self-regulation: The case of Facebook's content governance. *New Media & Society*. Advance online publication. https://doi.org/10.1177/1461444821989352

+ Show References

Google Scholar

Moschini I., Sindoni M. G. (2021). Language as the tip of the iceberg? Shedding a critical light on "hidden" discourse in digital platforms. *Discourse, Context & Media*, 42, Article 100505. https://doi.org/10.1016/j.dcm.2021.100505

Go To Reference

Google Scholar

Munn L. (2020). Red territory: Forging infrastructural power. *Territory, Politics, Governance*. Advance online publication. https://doi.org/10.1080/21622671.2020.1805353

+ Show References

Google Scholar

Musiani F. (2013). Governance by algorithms. *Internet Policy Review*, 2(3). https://doi.org/10.14763/2013.3.188

+ Show References

Google Scholar

Nieborg D. B., Helmond A. (2019). The political economy of Facebook's platformization in the mobile ecosystem: Facebook Messenger as a platform instance. *Media, Culture & Society*, 41(2), 196–218. https://doi.org/10.1177/0163443718818384

+ Show References

Google Scholar

Partin W. C. (2020). Bit by (Twitch) bit: "platform capture" and the evolution of digital platforms. *Social Media + Society*, 6(3), 1–12. https://doi.org/10.1177/2056305120933981

+ Show References

Google Scholar

Perriam J., Birkbak A., Freeman A. (2020). Digital methods in a post-API environment. *International Journal of Social Research Methodology*, 23, 277–290. https://doi.org/10.1080/13645579.2019.1682840

Go To Reference

Google Scholar

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. https://doi.org/10.1177/1461444816661553

Go To Reference

Google Scholar

Pridmore J. (2016). A social API for that: Market devices and the stabilization of digital identities. In van der Ploeg I., Pridmore J. (Eds.), *Digitizing identities: Doing identity in a networked world* (pp. 37–59). Routledge.

Go To Reference

Google Scholar

Pybus J., Coté M. (2021). Did you give permission? Datafication in the mobile ecosystem. *Information, Communication & Society*. Advance online publication. https://doi.org/10.1080/1369118X.2021.1877771

+ Show References

Google Scholar

Raetzsch C., Pereira G., Vestergaard L. S., Brynskov M. (2019). Weaving seams with data: Conceptualizing city APIs as elements of infrastructures. *Big Data & Society*, 6(1), 1–14. https://doi.org/10.1177/2053951719827619

Go To Reference

Google Scholar

Rieder B. (2013). Studying Facebook via data extraction: The Netvizz application. In *WebSci'13* (pp. 346–355). New York: ACM. https://doi.org/10.1145/2464464.2464475

Go To Reference

Google Scholar

Rieder B., Abdulla R., Poell T., Woltering R., Zack L. (2015). Data critique and analytical opportunities for very large Facebook Pages: Lessons learned from exploring "We are all Khaled Said." *Big Data & Society*, 2(2), 1–22. https://doi.org/10.1177/2053951715614980

Go To Reference

Google Scholar

er B., Hofmann J. (2020). Towards platform observability. *Internet Policy Review*, 9(4). https://doi.org/10.14763/2020.4.1535

+ Show References

Google Scholar

Schreieck M., Hein A., Wiesche M., Krcmar H. (2018). The challenge of governing digital platform ecosystems. In Linnhoff-Popien C., Schneider R., Zaddach M. (Eds.), *Digital marketplaces unleashed* (pp. 527–538). Springer. https://doi.org/10.1007/978-3-662-49275-8_47

+ Show References

Google Scholar

Snodgrass E., Soon W. (2019). API practices and paradigms: Exploring the protocological parameters of APIs as key facilitators of sociotechnical forms of exchange. *First Monday*, 24(2). https://doi.org/0.5210/fm.v24i2.9553

Go To Reference

Crossref

Google Scholar

Sohan S. M., Anslow C., Maurer F. (2015). A case study of web API evolution. In *2015 IEEE World Congress on Services* (pp. 245–252). New York: IEEE. https://doi.org/10.1109/SERVICES.2015.43

+ Show References

Google Scholar

Stratton C. (2020). Platform politics: Software as strategy in Apple's platform ecosystem. *First Monday*, 25(2). https://doi.org/10.5210/fm.v25i2.9948

+ Show References

gle Scholar

Tiwana A. (2014). *Platform ecosystems: Aligning architecture, governance, and strategy*. Morgan Kaufmann.

+ Show References

Crossref

Google Scholar

Tiwana A., Konsynski B., Bush A. A. (2010). Research Commentary—Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. *Information Systems Research*, 21(4), 675–687. https://doi.org/10.1287/isre.1100.0323

+ Show References

Google Scholar

van der Vlist F. N., Helmond A. (2021). How partners mediate platform power: Mapping business and data partnerships in the social media ecosystem. *Big Data & Society*, 8(1), 1–16. https://doi.org/10.1177/20539517211025061

+ Show References

Google Scholar

van Dijck J. (2021). Seeing the forest for the trees: Visualizing platformization and its governance. *New Media & Society*, 23, 2801–2819. https://doi.org/10.1177/1461444820940293

+ Show References

Google Scholar

van Dijck J., Poell T., de Waal M. (2018). *The platform society*. Oxford University Press. https://doi.org/10.1093/oso/9780190889760.001.0001

Go To Reference

Google Scholar

Venturini T., Rogers R. (2019). “API-based research” or how can digital sociology and journalism studies learn from the Facebook and Cambridge Analytica data breach. *Digital Journalism*, 7(4), 532–540. https://doi.org/10.1080/21670811.2019.1591927

Go To Reference

Google Scholar

Vis F. (2013). A critical reflection on Big Data: Considering APIs, researchers and tools as data makers. *First Monday*, 18(10). https://doi.org/10.5210/fm.v18i10.4878

Go To Reference

Google Scholar

Werning S. (2017). “Re-appropriating” Facebook? Web API mashups as collective cultural practice. *Digital Culture & Society*, 3(2), 183–204. https://doi.org/10.14361/dcs-2017-0211

Go To Reference

Google Scholar

Wilken R. (2014). Places nearby: Facebook as a location-based social media platform. *New Media & Society*, 16(7), 1087–1103. https://doi.org/10.1177/1461444814543997

Go To Reference

Google Scholar

Biographies

Fernando N. van der Vlist (MA University of Amsterdam) is a PhD candidate at Utrecht University and a research associate with the Collaborative Research Center “Media of Cooperation” at the University of Siegen. His research interests include digital methods and

internet studies, social media and platform studies, app studies, software studies, and data studies.

Anne Helmond (PhD University of Amsterdam) is an assistant professor of New Media and Digital Culture at the University of Amsterdam. Her research interests include software studies, platform studies, platformization, app studies, digital methods, and web history.

Marcus Burkhardt (PhD University of Giessen) is a postdoctoral researcher at the University of Siegen. His research interests include critical studies of algorithms, data, software and platforms, digital methods, cultures of coding as well as the history and theory of digital media.

Tatjana Seitz (MA University of Warwick) is a PhD candidate at the University of Siegen. Her research focuses on API studies at the intersection of economic, aesthetic, and data-driven concepts within the context of networked social interfaces.

Similar articles:

Open Access

The political economy of Facebook's platformization in the mobile ecosystem: Facebook Messenger as a platform instance

Show Details

Open Access

Weaving seams with data: Conceptualizing City APIs as elements of infrastructures

Show Details

Open Access

The Platformization of the Web: Making Web Data Platform Ready

Show Details

View More

ge recommends:

SAGE Knowledge
Book chapter

Sourcing Data from Social Media via APIs

Show Details

SAGE Knowledge
Book chapter

R for Social Media Analysis

Show Details

SAGE Research Methods

Collecting Data Using APIs Part 1: How to Understand APIs and Navigating API Documentation

Show Details

View More

Also from Sage

CQ Library

Elevating debate

Sage Data

Uncovering insight

Sage Business Cases

Shaping futures

Sage Campus

Unleashing potential

Sage Knowledge

Multimedia learning resources

Sage Research Methods

Supercharging research

Sage Video

Streaming knowledge

Technology from Sage

Library digital services