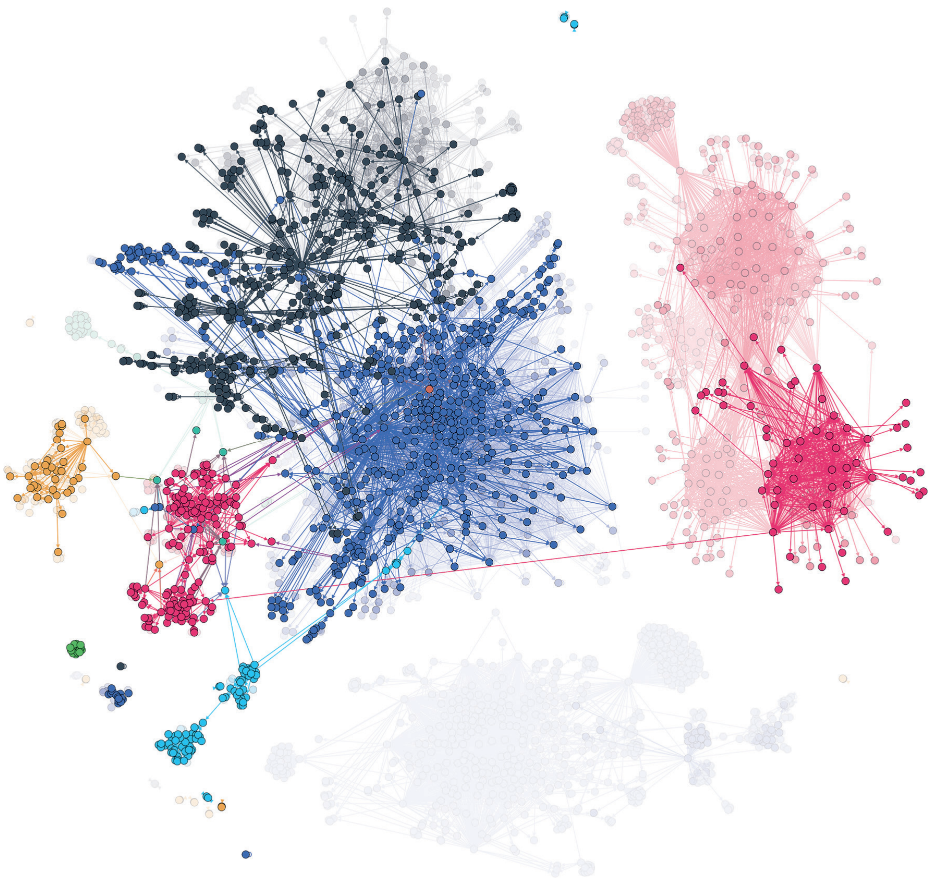
The Technicity of Platform Governance

Structure and Evolution of Facebook's APIs

Fernando N. van der Vlist[*,+], Anne Helmond[*,°],
Marcus Burkhardt[*], Tatjana Seitz[*]
[*]*University of Siegen*, [+]*Utrecht University*,
[°]*University of Amsterdam*



WORKING PAPER SERIES | NO. 20 | JULY 2021
Collaborative Research Center 1187 Media of Cooperation

Working Paper Series
Collaborative Research Center 1187 Media of Cooperation

Print-ISSN 2567–2509
Online-ISSN 2567–2517
DOI 10.25819/ubsi/9951
Handle: dspace.ub.uni-siegen.de/handle/ubsi/1938
URN urn:nbn:de:hbz:467-19380

Publication of the series is funded by the German Research Foundation (DFG).

Universität Siegen
SFB 1187 Medien der Kooperation
Herrengarten 3
57072 Siegen, Germany
www.sfb1187.uni-siegen.de
workingpaperseries@sfb1187.uni-siegen.de

The Technicity of Platform Governance
Structure and Evolution of Facebook's APIs

Fernando N. van der Vlist[*,+], Anne Helmond[*,°], Marcus Burkhardt[*], Tatjana Seitz[*]
[*]University of Siegen, [+]Utrecht University, [°]University of Amsterdam

Abstract: Researchers, policymakers, and competition and regulation authorities worldwide recognise application programming interfaces (APIs) for powering the digital economy and driving processes of datafication and platformisation. However, it is unclear how APIs tie into the power of, and governance by, large digital platforms. This article traces the relationality between Facebook's APIs, platform governance, and data strategy based on an empirical and evolutionary analysis. It examines a large corpus of (archived) developer pages and API reference documentation to determine the *technicity* of platform governance – the technical dimension and dynamics of how and what platforms like Facebook seek to govern. It traces how Facebook Platform evolved into a complex layered and interconnected *governance arrangement*, wherein technical API specifications serve to enforce (changes to) platform policy and (data) strategy. Finally, the article discusses the significance of this technicity in specifying the material conditions for app and business development 'on top of' platforms and for maintaining infrastructural and evolutive power over their ecosystems.

Keywords: APIs, platform governance, platform evolution, platform power, data strategy, Facebook

Introduction

Researchers, policymakers, and competition and regulation authorities worldwide recognise application programming interfaces (APIs) for their role in processes of datafication and platformisation; even as a way to 'dominate the digital world' (Iyer and Getchell, 2018; FT, 2020; van Dijck, 2020). 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 as the web became 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 that underpin today's vibrant platform ecosystems and the platform economies and societies they support. Consequently, Iyer and Getchell 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 insights with third parties. Competition authorities and regulators in Europe and in the USA increasingly scrutinise anti-competitive uses and potential data misuses centred around Facebook's APIs and the platform's monopoly power (CMA, 2020: Appendix J; FTC, 2020). In 2019, the FTC fined Facebook $5 billion for violating consumer's privacy rights by providing third-party developers access to friends data via its APIs and for not properly reviewing developers and their apps (FTC, 2019).

APIs enable programmatic communication and the exchange of data and functionalities between software systems. They power the digital platform economy and serve as the core elements of infrastructure that underpin the large ecosystems of apps and services (or 'complements') created by third parties and partners (or 'complementors'). Like software development kits (SDKs), 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 decentralisation of their services (Blanke and Pybus, 2020). Although APIs and SDKs 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 processes of datafication (e.g. programmatic data-sharing) and platformisation (Helmond, 2015; Pybus and Coté, 2021).

Despite broad recognition of their importance, we lack a comprehensive understanding of APIs as complex technical objects. As such, critical scholars argue that '[r]egulatory fixes require detailed insights into how technology and business models work' (van Dijck et al., 2018: 158) and call for the 'observability' of platforms as an explicit means of regulation (Rieder and Hofmann, 2020). We also know too little about the relationality between APIs and platform governance. Platform companies design and change their APIs to facilitate third-party app development in ways that influence – 'orchestrate' – the evolution of their ecosystems (Tiwana, 2014; 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 developers, businesses, digital marketers and advertisers, and academic researchers. In these environments, APIs enable and control the possible relationships and interactions between these different users and stakeholders in today's platform society and thus serve as a core technical dimension of 'platform governance' (Gorwa, 2019: 854). Accordingly, platform companies like Facebook and Twitter have responded to public controversy, criticism, and external social pressures not only with policy changes but also with changes to their APIs. After the Facebook–Cambridge Analytica (Fb–CA) data scandal, Facebook made changes to certain features, terms, and policies, and implemented restrictions on API data access and sharing (Barrett and Kreiss, 2019; Helmond et al., 2019).

This article presents the results of an empirically-informed 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 how and what 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 APIs for over a decade (Albright, 2018; Santos, 2019). Specifically, we consider the relationality between the design of Facebook's APIs, platform governance, and (data) strategy from a material-evolutionary perspective 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 Login, using current and archived Facebook developer pages. We provide original empirical materials for further historical platform research to better understand the evolutionary dynamics between API design and governance by platforms.

Building on prior research, the analysis details 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 (data) strategy. We thus contend that *governance by* platforms is about more than a platform's use or content moderation policy, terms and conditions, and corporate governance structure; instead, it is also the design (and redesign) of technical API specifications that condition and control the possibilities for the exchange of data and services between software systems and organisations. As such, this study contributes to the ongoing debate on 'platform governance' within the platform studies literature (e.g. Caplan and Gillespie, 2020; Gillespie, 2018; Gorwa, 2019; Gorwa et al., 2020; Medzini, 2020; Schreieck et al., 2018). We argue that it is important to study the technicity (and materiality) of governance by platforms like Facebook to understand the sources of their 'infrastructural power' (Blanke and Pybus, 2020).

In the next section, we provide an overview of the current literature on APIs and platform governance to position our contribution in these research areas. Second, we detail our material evolutionary approach to study Facebook's APIs against this background and describe our method of data collection as well as our (openly available) data set. Third, we describe the results of our empirically-informed case study of the structure and evolution of Facebook's API architecture. Finally, we discuss the relationality between API design, platform governance, and data strategy and the importance of a technical perspective on governance by platforms as a source of their power.

APIs Studies Meet Platform Governance

APIs have been studied by scholars across 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 contextualise our analysis of the technicity of platform governance.

First, APIs have been described as mechanisms of generativity and control. APIs enable third-party app developers to interact with a platform to access and exchange data and services through a standardised information exchange which ensures interoperability (Bodle, 2011; Tiwana, 2014: 7). They coordinate the 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' (Zittrain, 2008) by inviting third-party app developers to create new apps and services 'on top of' a platform using its APIs. This generative dimension of platforms has previously been understood as a form of participatory 'remix' or 'mashup' culture, as platform appropriation, and as value-adding activity (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 and Henfridsson, 2012). As such, APIs provide access to data and services in exchange for control (Eaton et al., 2015; Evans and Basole, 2016). Protocological technical objects like APIs serve as 'conduits for governance', or as 'artefacts of governance', where control is enabled on the level of API code (Bucher, 2013; Musiani, 2013). In fact, control over programming interfaces 'amounts to control over the platform and its evolution' and the platform's complements and complementors (Tiwana et al., 2010: 680). APIs thus facilitate infrastructural dependencies between platforms and apps, which we argue represents a source of infrastructural platform power, and provide 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 made their entry to promote a more 'social' experience of the web, new forms of API-based connectivity emerged to underpin today's 'data-intensive' web (Gerlitz and Helmond, 2013). As Langlois and Elmer suggest, platforms are increasingly 'weaving themselves in a new distributed infrastructure of life in all its forms' (Langlois and Elmer, 2019: 6). The widespread implementation of web APIs as the standard for data access and sharing has created new 'connected viewing environments' in the television industry (Lahey, 2016), and new kinds of data seams in the urban fabric of cities (Raetzsch, 2019). These developments centre on the role of APIs as the standard mechanism for interconnectivity, embeddedness, and scale growth but also raise concerns around power through platformisation (Blanke and Pybus, 2020; van Dijck, 2020) and 'infrastructuralisation', whereby platform-based services acquire characteristics of infrastructure (Helmond et al., 2019; Plantin et al., 2016). The outcome is that these new data-intensive fabrics are no longer open or public but instead are privatised and governed by platform companies.

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

Fourth, APIs have been pivotal to the business models and strategies of platforms and to the commercialisation of the internet in general. IS researchers have studied the economic and business dimensions of the API ecosystem as an API economy. In this economy, platform companies strategically provide data access through APIs to stimulate the development of 'API recombinations' to capture the value produced by these complements (Basole, 2016). APIs also facilitate the distributed capture of datafafied user engagement on third-party websites and apps, giving rise to web economies such as the 'Like economy' of the social web (Gerlitz and Helmond, 2013). The logic of participation thus became heavily commercialised. Others traced TripAdvisor's evolution into a diversified ecosystem of data-based services, where complementors assemble around new data forms to create additional services governed through APIs (Alaimo et al., 2020). Further, the strategic role of APIs in collecting new types of valuable data has been studied to understand the evolving data and business strategies of platforms (Wilken, 2014), by comparing available data points for online profiling (Bechmann, 2013), and by comparing API ecosystems (Evans and Basole, 2016). In short, APIs are not merely technical objects for software and app development but an integral part of a platform's data and business strategies.

Finally, APIs are commonly used and reflected upon as tools for academic research. Some researchers scrutinised the use of APIs for data collection purposes and the role of API-based research software tools as 'data makers' (Lomborg and Bechmann, 2014; Rieder, 2013;

Vis, 2013). Similarly, others considered the technicity of APIs as they intervene in empirical research by shaping their objects or phenomena of study (Puschmann and Ausserhofer, 2017; 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 But 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). In fact, 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 Facebook–Cambridge Analytica scandal and the subsequent 'APIcalypse' (Bruns, 2019), platforms like Facebook and Twitter severely restricted API data access and sharing. This impacted critical academic research into phenomena such as abuse, hate speech, and disinformation campaigns and as a result, the conditions of platform observability through APIs have worsened (Rieder and Hofmann, 2020). The way that platform APIs are governed raises questions around fair use and the need to look for alternative research methods suitable for a 'post-API' environment (Freelon, 2018; Perriam et al., 2019; Venturini and Rogers, 2019).

While these streams of API-related research all provide important insights into the politics of social media and platform APIs, they have not necessarily focused on the (evolutionary) dynamics between APIs as technical objects and the technicity, or the material conditions of how platforms have governed app and business development. On the one hand, media and communication scholars have focused on governance in relation to end-users and content, including platforms' policies and their terms and conditions (e.g. Caplan and Gillespie, 2020; Gillespie, 2018), the technical challenges and politics of algorithmic content moderation (Gorwa et al., 2020), and governance by algorithms (Musiani, 2013). Some considered platform governance in app development through discourse analyses of developer forums and technical documentation (e.g. Greene and Shilton, 2017; Moschini and Sindoni, 2021). On the other hand, IS researchers and software engineers have done empirical studies of APIs and their documentation to better understand structural platform changes, how these changes are communicated, or impact development (e.g. Medjaoui et al., 2018; Sohan et al., 2015). They also theorised how a platform's evolution is influenced by the coevolution of its architecture, its governance, and the 'environmental dynamics' of its ecosystem (Tiwana et al., 2010), and the challenges of governing ecosystems (Schreieck et al., 2018). We connect these research streams to highlight that platform governance is about use or content policy, terms and conditions, and guidelines as much as about API design and strategy.

Furthermore, our evolutionary perspective foregrounds a platform's capacity to *shape* – through governance and strategy – the evolution of the ecosystem around it in ways that impact outcomes (economic or otherwise). For instance through 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 end-users, developers, or businesses based on historical preference or use (e.g. Alaimo et al., 2020). Previous technical design choices, strategic decisions, and advantages often have an enduring influence into the present.

Studying the Material Conditions of App Development

Platforms like Facebook leave all kinds of material traces that document their operations that can be used to 'observe' platforms (Rieder and Hofmann, 2020) and reconstruct platform governance and data strategy as they have evolved. These material traces include information about the platform's APIs, SDKs, and other software tools on dedicated websites for developers (developers.facebook.com) and businesses (facebook.com/business). As these websites have been well-archived in web archives they are particularly suitable for empirical and historical platform studies (Helmond and van der Vlist, 2019). In fact, they are better documented than most because they serve as a platform's 'boundary resources' that assist third-party software developers (and business developers) in building, maintaining, and evolving apps and services (Ghazawneh and Henfridsson, 2012).

Boundary resources such as APIs and SDKs and their associated reference documentation 'expose and extend the platform architecture' (Dal Bianco et al., 2014: 13). This enables us to employ these materials to trace how a platform's architecture has changed or evolved over the years. Moreover, platform owners, complementors, and other actors collectively shape and reshape the evolution of a platform's boundary resources in a 'distributed tuning' process, showing 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 a 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 and Sindoni, 2021) and shaping the meaning of 'privacy' (Greene and Shilton, 2017). They provide important information and contain traces of *API governance* – that is, governance by platforms through their APIs. The amount of detail provided in the API reference documentation thus enables granular empirical analysis of the material conditions of third-party app development, how and why those conditions have changed or evolved, and the role of power therein.

The Facebook for Developers website covers the reference documentation for the entire Facebook Platform.2 This includes technical information about the APIs' architecture and structure, instructions on how to use them (e.g. how to read or write information to Facebook's social graph), and additional information about versioning, access levels, and rate limits, as well as specific data fields, edges, parameters, and permissions. Within the API reference documentation, Facebook currently refers to API objects as 'nodes' (e.g. the /user, /photo, /event, and /page nodes) or as 'endpoints'. 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 'user_location' permission is required to access 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).

Developers can use nodes and edges to access or exchange Facebook data and services. When developers request data from a node, it typically returns many additional details about that node (i.e. its properties), such as the 'created_time' and 'id' for a /post, up to as many as 51 details that give further context about that post. When they instead request data from an edge, it returns a list of the nodes connected to that specific edge (without all the additional details). Over the years, a complex layered structure of access controls, application permissions, and app review guidelines has evolved to govern and restrict API data access and sharing for most nodes and edges. Additionally, there are Facebook's Platform Terms and Developer Policies (FD-2021o; FD-2021a).

Archived Developer Pages and API Reference Documentation

This article's empirically-informed analysis is based on Facebook's developer pages as retrieved from the 'live' web and from the Internet Archive Wayback Machine. 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 Facebook API (beta) launch (August 2006 – February 2020).3 Because Facebook does not provide an archive of its developer website, these independent archived sources provide an important means for the observability of platforms. We combined multiple strategies to explore this large corpus of web sources because APIs are complex (and composite) technical objects that demand analysis at the different levels at which they occur and operate, like other types of software objects (Fuller, 2008). Therefore, we analyse the evolution of Facebook's APIs on the level of the entire API architecture, the level of individual API objects, and the level of application permissions.

On the 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. As such, the link structure embedded in the reference documentation reflects the API architecture. We derived and charted the link structure as it evolves with each new version. Additionally, we created a corpus of 178,972 web pages with annual 'snapshots' of archived URLs anywhere in Facebook for Developers to visualise the complexity and diversity of these APIs and to examine API naming conventions. We further analysed the associated 'changelogs' (FD-2021g; FD-2021c), 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 about 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.

On the level of individual API objects, we examined one of the core (and most connective) nodes in the entire reference documentation: the Graph API 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 central to Facebook's social network (as structured around people's user-profiles and friendship connections) and to 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 its descriptive properties and as a *relational object* in terms of its connections. The User and Page objects (the latter representing businesses, organisations, and public figures) are the two nodes that can authorise access tokens for apps to allow data access. Other data objects are typically linked to and 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. The same targeting options are available through Facebook's self-service advertising tools and enable us to examine how the targetable user has been governed through the MAPI (FB-2021).

Finally, we examined application permissions, which provide a way for apps to access data from Facebook (FD-2021l), and which 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. Until recently (GAPI v8.0), Facebook Plat-

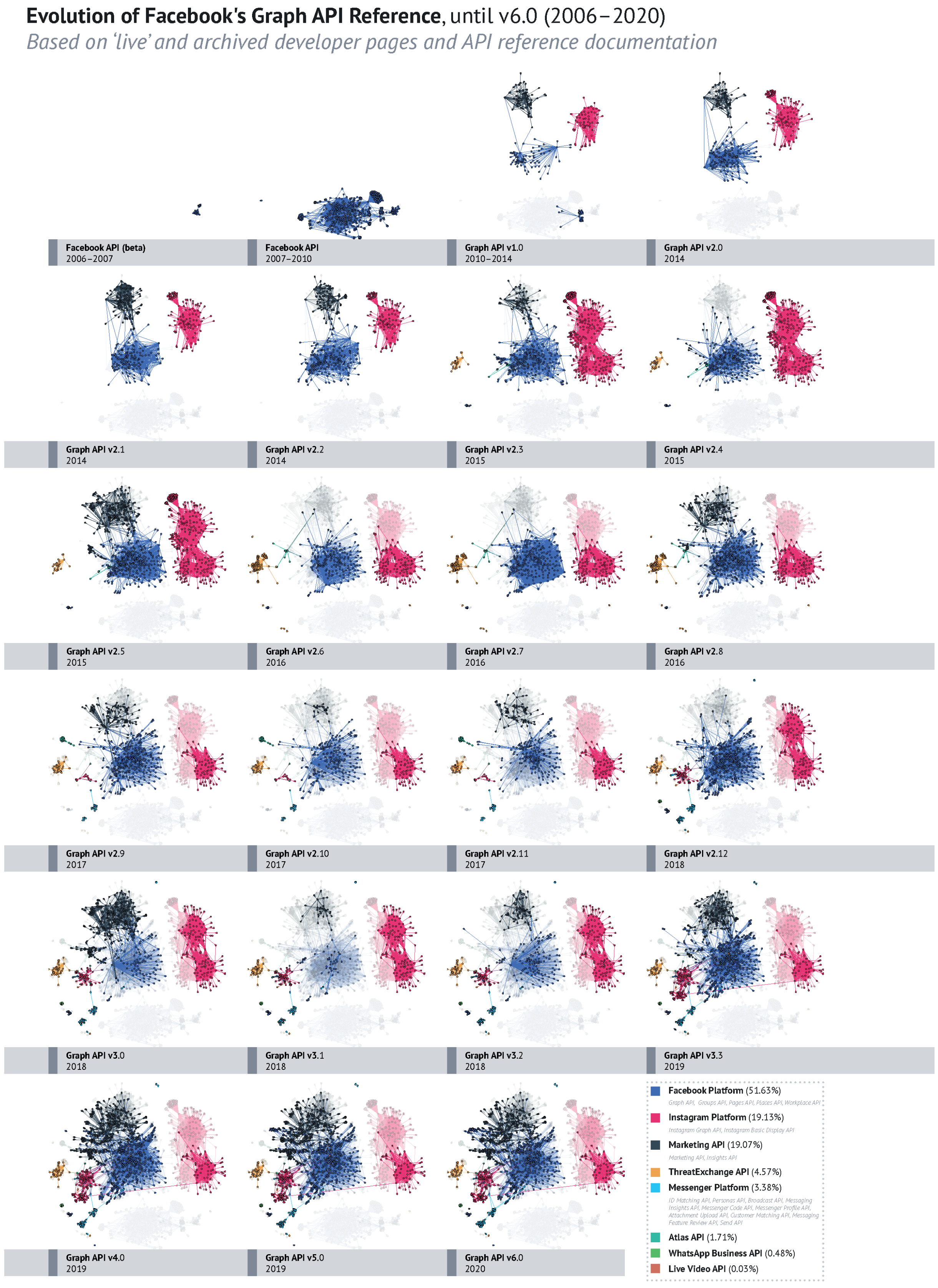

Evolution of Facebook's Graph API Reference, until v6.0 (2006–2020)
Based on 'live' and archived developer pages and API reference documentation
Facebook API (beta) 2006–2007
Facebook API 2007–2010
Graph API v1.0 2010–2014
Graph API v2.0 2014
Graph API v2.1 2014
Graph API v2.2 2014
Graph API v2.3 2015
Graph API v2.4 2015
Graph API v2.5 2015
Graph API v2.6 2016
Graph API v2.7 2016
Graph API v2.8 2016
Graph API v2.9 2017
Graph API v2.10 2017
Graph API v2.11 2017
Graph API v2.12 2018
Graph API v3.0 2018
Graph API v3.1 2018
Graph API v3.2 2018
Graph API v3.3 2019
Graph API v4.0 2019
Graph API v5.0 2019
Graph API v6.0 2020
🟦 **Facebook Platform** (51.63%)
Graph API, Groups API, Pages API, Places API, Workplace API
🟥 **Instagram Platform** (19.13%)
Instagram Graph API, Instagram Basic Display API
⬛ **Marketing API** (19.07%)
Marketing API, Insights API
🟧 **ThreatExchange API** (4.57%)
🟦 **Messenger Platform** (3.38%)
ID Matching API, Personas API, Broadcast API, Messaging Insights API, Messenger Code API, Messenger Profile API, Attachment Upload API, Customer Matching API, Messaging Feature Review API, Send API
🟩 **Atlas API** (1.71%)
🟩 **WhatsApp Business API** (0.48%)
🟥 **Live Video API** (0.03%)

Fig. 1: Evolution of Facebook's API Reference, until v6.0 (2006–2020) [small multiples]. Each tile represents the entire 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 rendered transparent in the background.

form distinguished between 'Basic' and 'Extended' Permissions for accessing Facebook data and services. Any app could by default access ('read') the data fields that belong to a User's 'public_profile', including their 'name' and 'picture'. When an app required access to additional data or to publish ('write') data to the platform, it needed to request extended permissions from the respective user(s) (FD-2008).

To contextualise observed changes on any level, 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 California state court (Appendix; FL-2019a). We thus draw on a variety of primary Facebook sources and external sources that provide important contextual information about specific changes. Finally, we used visualisations to support the analysis and communicate a sense of the complexity of Facebook's APIs and make our data sets openly available.

The Structure and Evolution of Facebook's APIs

This section presents the results of a multilevel analysis of the structure and evolution of Facebook's APIs. Taken together, these levels of analysis provide important insights about 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, 2014; 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 Platform's APIs, Instagram Platform's APIs, etc.). Initially (2006–2010), the platform only included the Facebook API, which provided data access related 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 methods, reflecting Facebook's evolution as a social network.4 With the launch of the current Graph API (2010), this API architecture style was redesigned on the logic of the (social) graph, which modelled Facebook's social network entirely in terms of 'nodes' (objects) and 'edges' (connections). Since then, the graph model has codified and represented any relationships between people and between people, objects, and activities on and off the platform (FD-2021h).

Core API Components: Cycles of Diversification and Integration

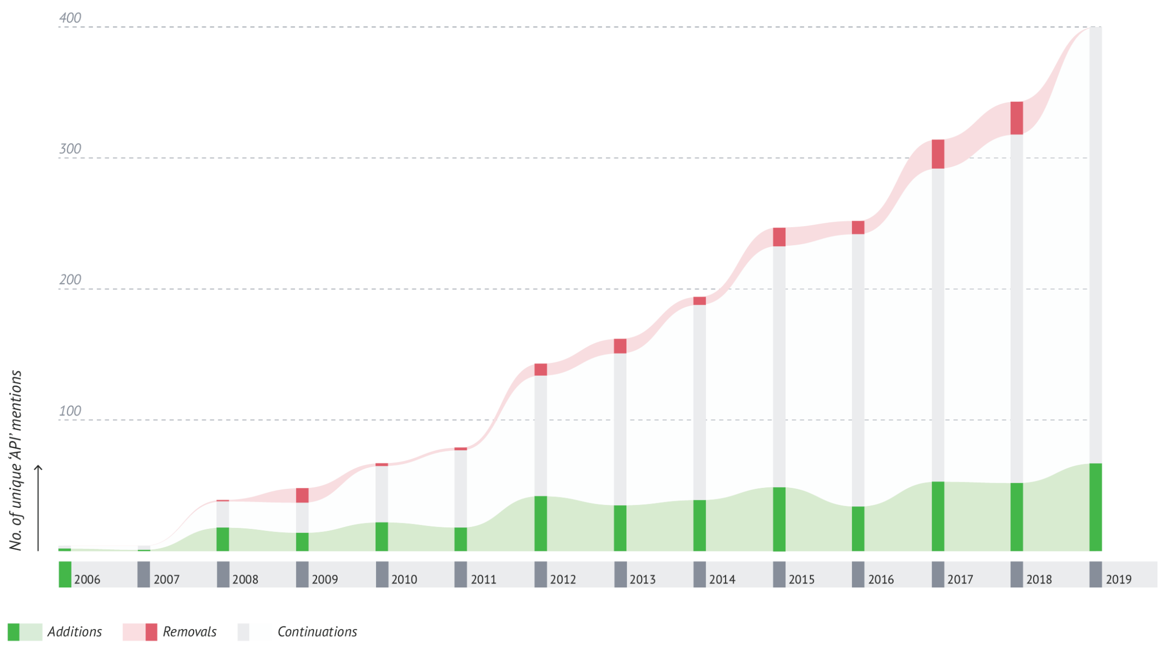
Since 2010, many new core APIs have been introduced and integrated, reflecting the different tools, products, and services that Facebook has created (e.g. the Marketing API, Messenger Platform, Workplace, etc.) or acquired (e.g. Instagram, WhatsApp, Atlas, etc.) over the years. Under each of these core API components of Facebook Platform, we find more specific APIs that provide access to specific data objects and functionality components. Figure 1 presents the evolution of the entire architecture of Facebook's APIs, including both these core components and specific endpoints, as well as their interrelations. This API architecture not only grew in size and complexity; it 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, platform governance is implemented through tiered API access models as well as through the ongoing (re)design and (re)structuring of API components.

Since 2014, the MAPI has increasingly merged (or technically integrated) with the GAPI. As such, Facebook's platform for advertising became part of its core development platform, rather than remaining a separate platform. While Facebook's own built and acquired platform products – Facebook's 'family of apps' – have remained separate for end-users, we see that they have long been interconnected in the back-end for business users. These changes were slowly rolled out over the course of several API versions. When Instagram (2012) and WhatsApp (2014) were acquired, their back-ends initially remained entirely separate from the rest of Facebook Platform. However, they later migrated to Facebook's data centres 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' (Isaac, 2019; IE-2014).

The longer-term evolutionary trajectory of Facebook's API architecture is characterised 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 entities referred to by Facebook as 'APIs' between 2006–2019. We identified 446 unique APIs in total. The vast majority encapsulate very specific GAPI and MAPI data and services that enable developers to build tools, products, and services more

Evolution of 'API' mentions within Facebook's Graph API Reference (2006 – 2019)
Based on 'live' and archived developer pages and API reference documentation



2006 Facebook API Generic API

2007 Facebook Platform API

2008 Action API Association Data Access API Association Data Definition API Batch API Batch Run API Batching API Client API Data API Data Store API Facebook Rest API Feed API Object Data Access API Object Data Definition API Permissions API Platform API Rest API SMS API User Preference API

2009 Application API Comments API Connect API Demographic Restrictions API Events API Facebook Platform Connect API Facebook Platform Rest API Facebook Restful API Open Stream API Page API Places API Restful API Search API Stats API

2010 Admin API Ads API Core API Dashboard API Facebook Status API Graph API Groups API Inbox API Insights API List API Marketplace API Messages API Migrations API Notifications API Open Graph API Platform Dialogs Graph API Real Time API Realtime API Rest Graph API String API Subscriptions API Time API

2011 Async API Backend API Backwards API Chat API Credits API Credits Graph API Dialog API Frontend API Games API Marketing API Messaging API Monetization API Page Graph API Payment API Requests API Scores API Sponsored Stories API Thread Graph API

2012 Achievements API Ads Preview API Ads Rest API Asynchronous API Audio API Batch Graph API Camera API Client Side API Credits Order API Dashboard Rest API Facebook Scores API File API Fql API Graph Batch API Graph Rest API Group Graph API High Score API Linter API Login API Medium Graph API Medium Insights API Message Graph API Milestone API Mobile Device API Navigation Timing API Network Information API Offers API Open Graph Photos API Open Graph Search API Orders API Page Post Visibility API Payer Promotion API Payments Reporting API Post API Real Time Updates API Reporting API Reports API Sponsored Results API Subject Graph API Test User API Time Updates API Updates API

2013 A/B Testing API Access Tokens All Graph API Ad Account API Apps API Domain API Friends API Game Groups API Gamorlive API Graph Insights API Id API Insights Graph API Message API Mobile API Object API Page Locations API Pages Marketing API Parse API Pay Dialog API Payment Graph API Places Search API Preferred Currency API Preview API Profile API Public Feed API Push Status API Response API Role API Send API Shortcut API Stream API Tag API Trial Pay API Trial Pay Offers API Upload API Video API

2014 Ad Report Stats API Ads Insights API Ads Reporting API App Events API App Insights API App Links Hosting API App Links Index API Atlas API Audience Network Native API Behaviors API Business API Business Mapping API Conversion Pixel API Custom Audiences API Disputes API Facebook Tag API Friends Graph API Games Achievements API Geo Targeting API Hashtag Counter API Hosting API Index API Instagram API Invitable Friends API Inviting API Keyword Insights API Mobile Hosting API Mobile Measurement API Native Ads API Page Feed API Place Picker API Social Context API Taggable Friends API Tagged Places API Topic Feed API Topic Insights API Topic Search API Trending API Trends API

2015 Ad Label API Ad Tag API Ads Activity Log API Ads Management API Aggregated Instant Articles API App Development API App Link API App Link Index API Audience Management API Audience Network Reporting API Business Manager API Buying API Custom Audience Pixel API Custom Audience Pixel Stats API Data Model API Device Login API Embedded Video Player API Game Groups Tutorials API Game Request API Game Services Tutorials API Geocoding API Inline Page Post API Insights Page API Instagram Ads API Instant Articles API Instant Articles Insights API Labels API Mapping API Measurement API Media Solutions API Minimum API Mutual Friends API Page Insights API Page Videos API Pixel Stats API Prediction API Reach And Frequency API Reach Estimate API Reachestimate API Refunds API Report Stats API Share API Sharekit API Showcase API Source API Story API Targeting Search API ThreatExchange API Xmpp Chat API

2016 Account Management API Accountkit API App Events Export API Asset Management API Audience Network Native Ad API Business Asset Management API Business Management API Catalog API Crossposted Videos API Device Share API Devices API Dynamic Post API ID Graph API Instagram Ads Post Moderation API Lead Ads API Live API Live Video API Messenger API Messenger Platform API Offline Conversions API Page Instagram Accounts API Product Catalog Preferences API Reactions API Recommendations API Rights Management API Rights Manager API Send/Receive API Three Degree Mutual Friends API User Profile API Video Edit API Video Upload API Webhooks API Wit.Ai API Workplace Account Management API

2017 Account Preferences API Accountkit Graph API Accountkit Rest API Actiontag API Ad Copy API Advanced Measurement API Advertiser API App Marketing API Atlas Reporting API Attachment Upload API Audience API Audience Create API Audience Network API Campaign API Checks And Quality API Checks API Company API Conversations API Crowdtangle API Current Place API Customer Matching API Detailed Targeting API Export API Games Ads API Id Matching API Instagram Platform API Instagram Stories Ads API Jobs API Marekting API Marking API Messaging Insights API Messenger Code API Messenger Profile API Page Messaging Insights API Pass Thread Control API Paymentrequest API Place Information API Placement API Places Graph API Preferences API Product Catalog API Product Feed Rule API Product Feed Rules API Product List API Publisher API Quality API Resumable Upload API Rules Engine API Split Testing API Suggested Rules API Take Thread Control API Thread Settings API Workplace API

2018 3D Posts API Add Instagram API Ads Marketing API Broadcast API Business Discovery API Business Management Asset API Catalog Batch API Certificate Transparency API Collection API Comment Moderation API Contacts API Content Publishing API Conversation Graph API Current Place Feedback API Destinations Marketing API Facebook Live API Game Switch API Handover Protocol API Hotel API Hotel Room API Hotel Rooms Batch API Information API Instagram Graph API Instant Articles Transformer API Instant Games API Lead Ads Retrieval API List Batch API Media Solutions Live API Media Upload API Mentions API Messaging Feature Review API Messenger Expressions API Messenger Send API NLP API Page Change Proposals API Page Scoped Id API Page Upcoming Changes API Pages Categories API Payment Dispute API Payment Request API Psid API Public Figure API Query Event API Register Account API Rules API Shops API Sponsored Message Ads API Store Visits Custom Audience Eligibility API Thread Control API Thread Owner API Whatsapp Business API Workplace Graph API

2019 Ad Creative API Ad Keyword Stats API Ad Library API Ad Rules API Ad Video Library API Ad Volume API Adcampaign API Ads Insights Edge API Ads Manager App API Basic Display API Bot Subscription API Business Onboarding API Canvas Ads API Canvas API Categories Management API Commerce API Commerce Payout API Commerce Platform API Connected Player API Credit Allocation API Custom Labels API Custom Update API Direct Feed Upload API Direct Upload API Document Interaction API Dynamic Ads API Finance API Fundraiser API Generation API Hashtag API Hashtag Search API Home Screen Shortcut API Instagram Basic Display API Instagram Content Publishing API Lead Retrieval API Leaderboard API Leadgen API Leads API Live Polls API Managing Pages API Marketing Leads API Matchmaking API Messenger User Profile API Onboarding Clients At Scale API One Time Notification API Order Management API Page About Story API Page Messaging API Particle System API Persona API Player Stats API Product Catalog Batch API Product Feed API Real Time Batch API Remaining Ads API Send And Receive API Sequence Sampler API Split Testing And Lift API Static Recommendations API Store Visits API Subscriptions Syncing API Support API Targeting API Tax Override API Whatsapp API Whatsapp Business Management API Workplace Live API

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

securely and efficiently. These special-purpose APIs further serve to promote particular use cases for app and business developers. As such, the developer pages do not only serve as technical reference documentation but they also have a communicative function for developers by signalling use cases (cf. Dal Bianco et al., 2014; Ghazawneh and Henfridsson, 2012). 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. Furthermore, while core APIs accommodate the broad range of users and uses, the additional specific-purpose APIs enable targeted governance of these users and uses at the level of specific API endpoints.

Changelog: The Transition to a Stable Platform

Developers require stable platforms 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 industries and societal domains.

At first, Facebook did not systematically communicate about API changes; instead, it only wrote about some API updates on its Developer News page (FD-2006a). This is reflected in Facebook's internal motto to 'move fast and break things', which also impacted dependent app developers. Consequently, due to mounting criticism from developers, Facebook started publishing a Developer Roadmap (2010) in a 'spirit of openness and transparency' and 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 those developers who at the time requested more frequent updates to the API reference documentation to increase the reliability of the platform (FD-2010a). The operation led to improvements in the reference documentation, bug fixes, and stability improvements by introducing a 'breaking change policy' (FD-2011a; FD-2011b). Communication about upcoming changes is important, especially when these concern changes that break dependent apps and services. After that, Facebook started posting weekly updates about any platform changes on its Developer Blog (formerly Developer News), marking the transition towards 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 minimised development risks for complementors, particularly businesses.

With the release of GAPI v2.0 (2014), Facebook made a number of key changes to announce and document API changes: It introduced API versioning (and retrospective version numbers) for the core GAPI to communicate about upcoming changes, as well as a two-year transition period and stability guarantee for developers to serve a 'more stable platform' and provide ample time to address upcoming 'breaking changes'. 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, 2014). The platform further introduced versioning to the Ads API (now Marketing API) and aligned its versioning and release cycles with the core GAPI soon after (FD-2014b).

This marked an important change from a continuous development and release cycle (accompanied by unpredictable changes) to scheduled and versioned release cycles, and it also standardised communications between Facebook and third-party developers through developer pages and reference documentation. From that moment onwards, changes to the GAPI and MAPI have been documented in the changelog because of its important communicative function: It informs developers (and other complementors) about how and when they should update their tools, products, and services to upgrade and comply with the new released API version (along with an Upgrade Guide). 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 ecosystem of apps and services.

Figure 3 shows the evolution of Facebook's APIs as documented in the changelog going back to v2.1 (2014). It documents any additions of new features, changes, deprecations, as well as the introduction of App Review requirements for certain API endpoints, affected nodes (or their fields, edges, parameters, permissions, etc.), and affected API methods (for reading, creating, updating, or deleting Facebook data). Additionally, this changelog is useful to determine the temporality of API evolution and governance.

Between 2014–2017 (GAPI v2.1–v2.4), there was a professionalisation and commercialisation of Facebook's business side. We see this with announcements and scheduled releases related to the MAPI for business developers. There were many additions to the MAPI in this period, which significantly expanded the data and services accessible to the platform's business developers and marketing partners. Further, we see how other new components, such as Messenger Platform, are gradually included in the GAPI Changelog, reflecting their integration with Facebook's core technical platform.

In mid-2017, there were many deprecations related to the MAPI (e.g. fields, permissions, targeting options, etc.). Facebook introduced additional restrictions on the GAPI and MAPI, and deprecated legacy APIs to improve data protection and permission requirements in the

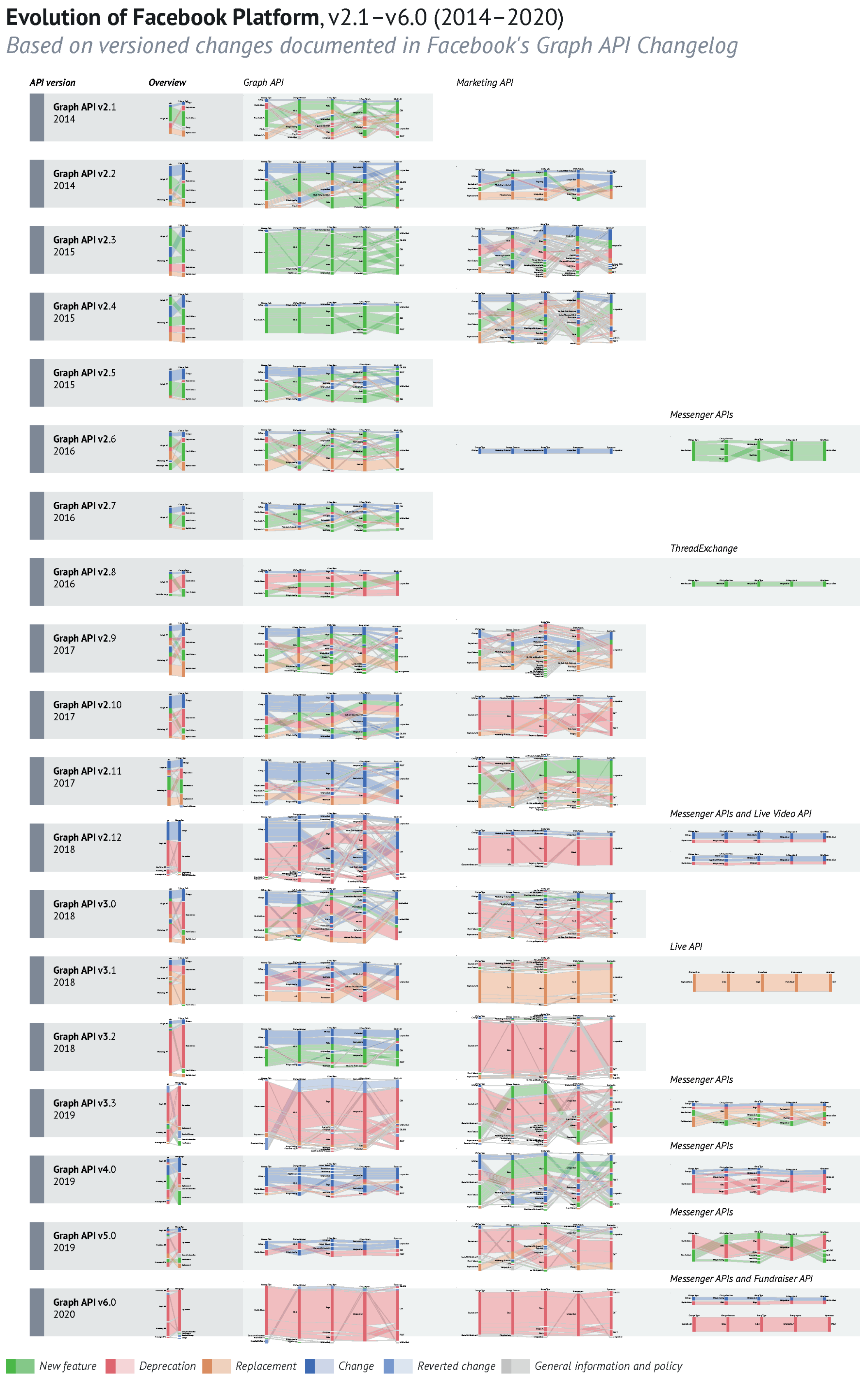


Fig. 3: Evolution of the Facebook Graph API Changelog, v2.1–v6.0 (2014–2020) [small multiples]. The changelog documents any versioned changes to the Graph and Marketing APIs,5 and any of Facebook's own products or services that rely on them.

wake of public controversies and criticisms about the platform's role regarding the 2016 UK EU-membership referendum, the subsequent US presidential election, and the Fb–CA data scandal (FD-2018c).6 Additionally, the EU General Data Protection Regulation (GDPR) came into force in May 2018, imposing stricter data protection obligations onto Facebook and other firms and organisations 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 (e.g. to data fields and permissions) but did not alter the internal structure of the API architecture, attesting to the robustness and adaptive capacity of the platform architecture style. Many of these changes ended up being '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. Because these breaking changes take effect immediately, they require urgent action by developers of apps relying on the respective API endpoints. 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 relation to application permissions.

(2) 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'). Additionally, we see the consequences of this in terms of how data objects are embedded in Facebook's graph-based data model, data-sharing with complementors, and their relationship to the platform's data strategy. Object-level API design decisions underpin Facebook Platform and any of its 'platform instances' (e.g. Instagram, WhatsApp, Messenger, and Workplace) because they have been integrated into a unified data infrastructure (Nieborg and Helmond, 2019). Moreover, these design decisions impact the platform's business side because a data object's fields and edges also serve as targeting options for Facebook's suite of (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 2006–2020. Between 2006–2010 (until GAPI v1.0), the User is one of the 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.

The release of GAPI v1.0 (2010–2014) marked a turning point for the platform's data structure because Facebook Platform was restructured according to the logic of the (social) graph. With the new graph-based data model, data objects came to be defined by their connections to other data objects and thus formed relationship networks. The User's fields (properties) are mostly defined by the user itself, while its edges (connections) emerge through the user's online activities, behaviours, 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. Further, the new graph-based data model also impacted app development and data use (i.e. data access and sharing). The new data model represented Facebook's vision of a 'social' web where its users are connected to each other, to other Facebook data objects, and to data objects outside of the platform's boundaries. Facebook appealed to third-party app developers to implement its social buttons on their websites at this time (Gerlitz and Helmond, 2013) and released the Open Graph protocol (2010) in an effort to standardise data formats on the web. This was a strategic move that helped to make a wealth of external (i.e. unstructured) data sources 'platform ready' (Helmond, 2015) and integrate those data sources with Facebook's data infrastructure. In short, 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 Facebook's own social network.

Most changes to the User object between 2014–2018 (GAPI v2.0–v2.12) were minor changes, 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). Additionally, 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 complex *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 becomes the central gateway through which all of the user's roles are governed by the platform.

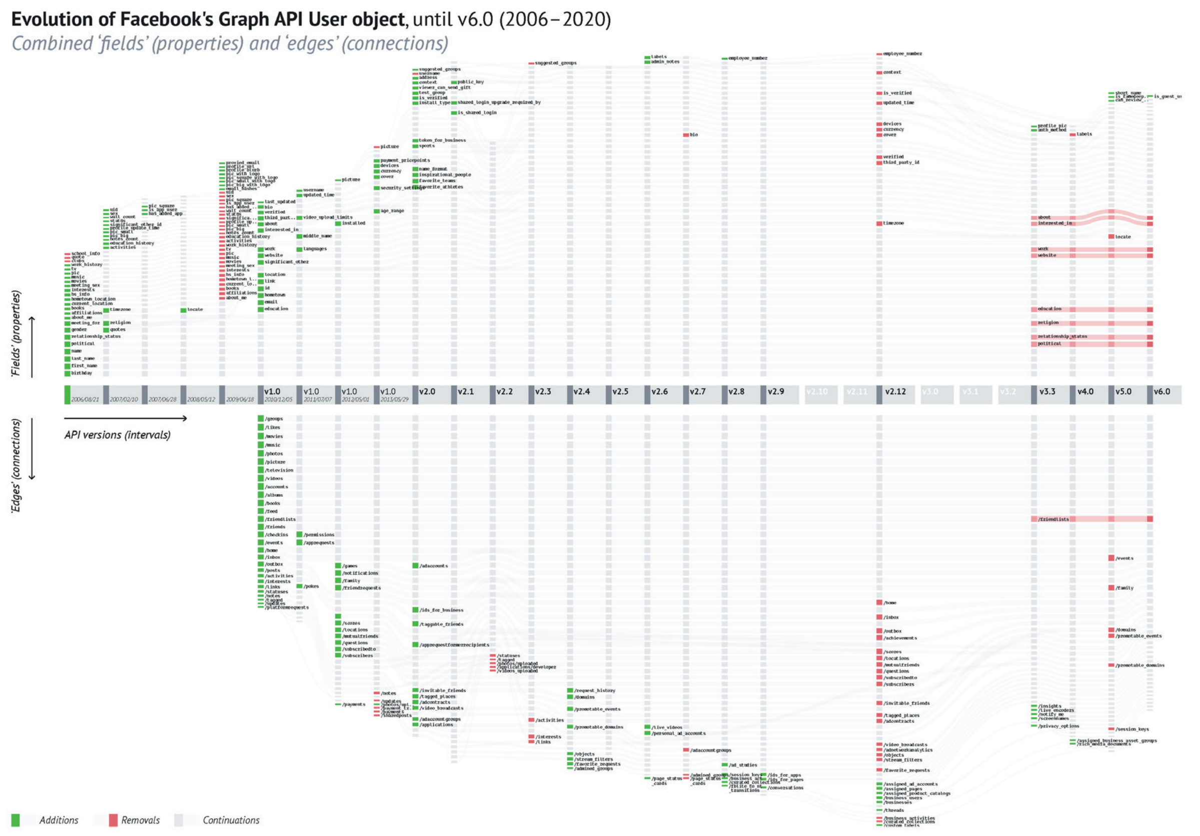


Fig. 4: Evolution of the Facebook Graph API User object, until v6.0 (2006–2020). The User object represents a user on Facebook as a combination of properties ('fields') and connections ('edges'). Each vertical slice displays the fields and edges of the User object for one API version.

In some cases, API objects are governed on the microscopic level of data fields. In early 2018, we observe that the User's 'interested_in' field is no longer available in the MAPI for targeting of people in France (FD-2016), then in Europe 'due to local laws' (FD-2017c), and then 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 and 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 organisations (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 organisations that housing, employment, and credit ads became a 'special ad category' with fewer available targeting options in compliance with US non-discrimination laws (FNe-2019a; FP-2021b; FD-2021k). Nonetheless, a year later, nonprofit newsroom *The Markup* reported that targeting 'multicultural affinity categories' was still possible, after which Facebook announced it removed them in the name of 'simplifying and streamlining our targeting options' (FB-2020). In short, we also witness how the User, as a *targetable user*, has evolved within the MAPI reference documentation in response to external social and legal pressures.

GAPI v3.0 (April 2018) was the first version to implement major changes on the User object level in the wake of the Fb–CA data scandal disclosed in March 2018. Facebook deprecated many 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 the users via MAPI-endpoints.

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

(3) Application Permissions Level: Facebook Login as a Control Point

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 involves distinct access controls and privileges (e.g. read-only access, read/write access, etc.) for different app and user types. The majority of application permissions are now requested through Facebook Login (Figure 5).

Permissions for applications did not exist until 2008. Instead, developers had access by default to 'your profile info (excluding contact info), your photos, your events, and most importantly, your friends' (FD-2006b). If Facebook users did not want an app – or 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 developers (or application) side with opt-in permission requests. Extended permissions 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 now needed 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 by default through the API, 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 permissions dialogue (FD-2010c; FD-2010e). Consequently, it is 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 your application' (FD-2010b). In the new permissions model, apps could access friends' basic profile information via the User object, without explicit permission from each of their friends, while 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, permissions now controlled on an *individual level* which apps could read or write data to the platform and which apps could access user and friends data. The increasing number of data fields associated with the User object in this period, including new Open Graph API actions, led to a sharp increase in the number of extended permissions – from eight to 49 (2008–2010) to 72 (May 2012). Between 2011–2012, 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.

There were several notable changes between 2014–2018 (GAPI v2.0–v2.12). First, the platform restricted access to users' friends' data in response to mounting concerns about users' data and privacy. The friend list no longer belonged to the basic permissions and now required apps to request extended permission from each app user (FD-2014c). Additionally, whereas the GAPI User Friends (/user/friends) endpoint in v1.0 returned lists of users' friends, v2.0 only returned lists of friends who also installed the app and gave 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 App Review as well. Facebook informed the UK Competition and Markets Authority (CMA) that this was

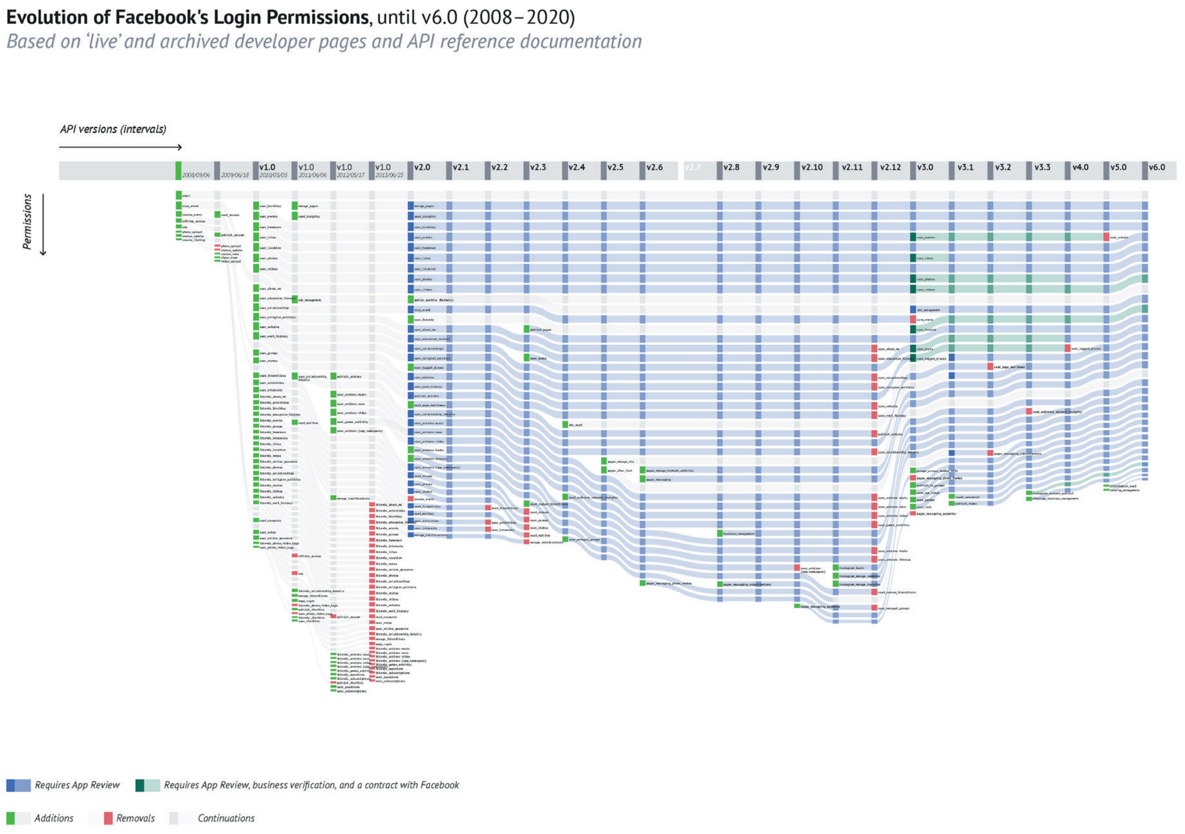


Fig. 5: Evolution of Facebook Login Permissions, until v6.0 (2008–2020). Permissions provide a way for applications to access data from Facebook and the largest number of permissions is requested through Facebook Login. Each vertical slice displays the permissions requested through Facebook Login for one API version.

'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. Further, the Login Review process launched to ensure that apps request only those permissions they need (FD-2014a; FD-2021f; FD-2021e). As such, Facebook Login is now used for user *authentication* (i.e. signing in) as well as for *authorisation* (i.e. handling permission requests 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 apps submitted for review would add value to the platform's ecosystem. In fact, 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 friends 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' (apps were expected to 'take data, give data') (FL-2019e). In short, there were now two competing accounts that explained these API changes: On the one hand, Facebook tells a story where 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. Or, 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). Further, an increased number of permissions (e.g. 'user_friends', 'user_likes', and 'user_photos') were 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). Further, more permissions now required App Review as well as requiring developers to verify their accounts for any apps that request access to that data (e.g. 'user_friends', 'user_likes', and 'user_photos') (FD-2018d; FD-2018f; FD-2021b). Facebook Login received its own Changelog to document changes to permissions as it had become the core authentication service for both end-users and developers as well as 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-2018g; FD-2018e). Additionally, the tiered MAPI access structure for partners and businesses was simplified (FD-2018b). In short, while it may appear as if Facebook Platform only grew or expanded over the years, it has also restructured at crucial moments in (partial) response to external social and legal 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. There is no longer a distinction between basic and extended permissions; instead, all permissions other than email 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 to basic profile information, photos, and videos need to request separate Instagram Permissions (FD-2021m). Examples of such apps include popular photo and video editing (mobile) apps and apps for exporting or printing users' photos (Gerlitz et al., 2019). The distinct treatment – governance – of these app types, we suggest, reflects Facebook's dual governance strategy for business users and creators on the one hand and for end-users on the other hand.

Conclusion: Governance by APIs

This article examined the relationality between Facebook's APIs and governance by the platform from an empirically-informed evolutionary perspective. We traced how Facebook's APIs have evolved from a simple programming interface for development into a complex layered and interconnected governance arrangement. Technical API specifications provide a plethora of governance and control mechanisms and serve to enforce (changes to) platform policy and (data) strategy, especially for app and business development.

The analysis highlights the technical dimension and dynamics of governance by platforms and how a platform's evolution is linked to – that is, *coevolves* with – strategic decisions about platform governance and data access within the platform's ecosystem. We emphasise that this technicity of platform governance facilitates the strategic orchestration of (asymmetric) relationships with complementors (Eaton et al., 2015; Tiwana, 2014; Tiwana et al., 2010). In our case study of Facebook's APIs, this technicity articulates governance and control mechanisms on several platform levels, comprising what Caplan and Gillespie conceptualise as a 'tiered governance' strategy, wherein different users face different rules, material resources, and procedures (2020: 6). However, whereas they focused on YouTube's governance mechanisms for creators (the YouTube Partner Program, content moderation, and demonetisation), this article examined Facebook's API governance for app and business development. Examining this technical dimension of governance by platforms requires distinct methods and materials. As a result, this study has different implications for our understanding of platform governance and power.

In this last section, we discuss the significance of the technicity of platform governance – *governance by* APIs – first from a material perspective and then from a strategic perspective. We underscore that APIs simultaneously establish the material conditions for the app and business development 'on top of' platforms and for gaining infrastructural power (cf. Blanke and Pybus, 2020; Eaton et al., 2015). Despite its significance, infrastructural power remains one of the blindspots in the European policy debate on platform power (Busch and Graef, 2021).

From a material perspective, API governance targets the conduct of a platform's complements (i.e. apps and services) and/or its complementors (i.e. developers, businesses, advertisers, etc.) operating within the platform's ecosystem. As such, the analysis augments the current literature on platform governance, which is primarily focused on governance and power over the conduct of end-users, creators, and content producers (e.g. Caplan and Gillespie, 2020; Gillespie, 2018; Gorwa, 2019; Gorwa et al., 2020; Medzini, 2020). We contend that boundary resources like APIs and SDKs are a critical part of a platform's governance mechanisms because they constitute the material infrastructures of platforms. Crucially, they enable complementors to create apps and services (ecosystems) that enlarge a platform's scale and scope while also increasing its infrastructural power (Blanke and Pybus, 2020; Gerlitz et al., 2019; Helmond et al., 2019; Pybus and Coté, 2021; van der Vlist and Helmond, 2021).

Our analysis shows that Facebook's APIs have become a complex layered arrangement of technical objects that evolve and govern in different ways. In this governance arrangement, Facebook's application permissions, Facebook Login, and App Review have become increasingly important as the primary mechanisms for governing data access and sharing. As such, application permissions enable distinct and approved data transactions (Pybus and Coté, 2021) as well as govern the (automated) data-sharing relations between a platform, its complementors, and its end-users. Moreover, the additional required App Review and Verification governance layers served to block bad actors and the entry of competitors.

We argue that large platforms like Facebook have *evolutive power* insofar they can influence or shape the evolution of their platform as well as the ecosystem of apps and services that have been built – and depend – upon its APIs. That is, large platforms have the ability (e.g. the technical and financial means) to establish and govern the material conditions under which certain forms of participation in the platform's ecosystem are allowed, sustained, and can thrive. An analysis of evolutive power thus captures a platform's ability to 'orchestrate' its ecosystem evolution, its relationships with complementors (i.e. its power over complementors), and the relationships among those complementors as mediated by the platform (i.e. its intermediary power). When platforms like Facebook make changes to its APIs, it causes large ripple effects throughout the entire platform's ecosystem (and market) and impacts the tools, products that developers, businesses, academic researchers, or others have put in place. For example, Facebook's API changes to access friends' data in 2014–2015 severely impacted the business models and apps of complementors as well as academic research tools, causing shutdowns across the entire ecosystem of apps and services that relied on this data (e.g. Constine, 2015; FL-2019b).

Platforms' application and development boundary resources play a central role in this because they establish the material conditions of participation and control (Dal Bianco et al., 2014: 13; Eaton et al., 2015). API architecture design shapes, on an infrastructural level, what is technically feasible on a given platform, while a platform's governance shapes what is allowed, encouraged, or technically and economically viable within that platform's ecosystem. Facebook Platform may allow diverse user and stakeholder groups to participate in its ecosystem but 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: 219). In the case of Facebook, we see that certain uses of Facebook data and services have been allowed and encouraged (e.g. building 'rich social apps'), whereas other uses have been discouraged or restricted (e.g. 'data export tools').

At the same time, Facebook has faced increasing external social pressures as well as legal requirements around data protection and privacy, leading to significant changes. However, while some of these API changes may have improved data protection and privacy controls for end-users, they typically also serve strategic purposes (e.g. to limit or restrict data access by selected competitors). As such, the 'tuning' of boundary resources may be distributed (shaped by 'a network of heterogeneous actors and artifacts', including external social pressures) but the ability to influence the tuning process is also asymmetric (cf. Eaton et al., 2015; Gerlitz et al., 2019).

From a strategic perspective, Facebook Platform has undergone several cycles of *diversification* with the continuous addition of new API endpoints and *integration* of new API components into its core technical platform. Many of these new API components and endpoints originated from internal development at Facebook (e.g. Facebook Messenger) while others originated from successful mergers and acquisitions (e.g. Instagram, WhatsApp, Atlas Solutions, LiveRail, etc.) or from Facebook's marketing partnership strategy (e.g. the diversification of Marketing API endpoints). This ongoing diversification of API endpoints and integration of API components has facilitated what Blanke and Pybus describe as the decentralisation of platforms into 'service assemblages', which has enabled large platforms like Facebook and Google 'to shift the dynamics of competition and monopolization in their favor' (Blanke and Pybus, 2020: 2; cf. Pybus and Coté, 2021). Strategies of diversification and integration are related; both are important for understanding the infrastructural power of these platforms.

In the case of Facebook's evolution, we see that its diversification and integration strategies have coevolved. The aforementioned acquisitions by Facebook were all, eventually, integrated into Facebook's core technical platform (i.e. the Graph API), although the process took several years. Moreover, these integrations initially occur in the back-end first (or only), enabling complementors to reap the benefits of consolidated platforms. As such, Facebook has maintained a fragmented front-end for end-users: It offers not one single but a 'family of apps', where each app speaks to a different segment of the platform's user population or accommodates a different set of user practices (Nieborg and Helmond, 2019). Many users are unaware that Facebook also owns Instagram, WhatsApp, and other popular apps, although the November 2019 'from FACEBOOK' (*sic*) company rebranding is clearer about Facebook's ownership of all these apps (Constine, 2019; FNe-2019b).

Platform-level integrations like these are not only technical but also strategic processes that contribute towards the consolidation of platforms' data assets and power (cf. Gawer, 2020). Consolidation enables platform owners to leverage, as well as *strengthen* the indirect (or 'cross-side') network effects of their platforms even more effectively. APIs facilitate the exchange of data and services between the platform and each of these sides. The stronger the network effect, the larger the value provided and captured by the platform owner, driving even greater reach and scope for the platform, more accurate content or friend suggestions, and raised barriers to entrants. Moreover, these platform-level integrations likely make it more difficult to split up platforms like Facebook because of the added value – or extended capabilities – for other user groups. More generally, then, we contend that these strategies of diversification and integration are driving the evolutionary process of 'platform capture', as conceptualised by Partin (2020), wherein the technical architecture of a platform evolves through the leveraging ('exploitation') of power asymmetries in its relationships with complementors ('dependents'). As such, diversification and integration contribute to platforms' infrastructural and evolutive power.

To truly understand how the relationship between a platform like Facebook and its ecosystem is governed, it is important to track changes in the dynamics between a platform's API design (or technical architecture), governance, and strategy in relation to public controversy, or other social and legal pressure to change. We recommend future research on the technicity of platform governance to explore the material conditions and evolutionary dynamics of APIs as (complex) artefacts of governance through (comparative) case studies of other digital platforms. Additionally, more research is needed on how changes to popular APIs may cause ripple effects (and potential breakdowns) throughout the interconnected platform ecosystem. In this, web archives can play an important role in providing a means for the observability of platforms by preserving the material traces of platform evolution.

Acknowledgements: Thanks to Angeles Briones (DensityDesign Lab, Polytechnic University of Milan) for information visualisation assistance during the research process, and the Center for Advanced Internet Studies (CAIS) for funding and hosting our working group.

Data availability: The data that support the findings of this study will be made openly available in the Open Science Framework (OSF) at https://doi.org/10.17605/osf.io/wfxyp.

Funding: This work was supported by the German Research Foundation (*Deutsche Forschungsgemeinschaft*, DFG), project number 262513311 (SFB 1187: 'Media of Cooperation'); and the research programme 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).

Notes

1. Although Facebook did launch API-based initiatives to build academic partnerships and improve transparency, including SOCIAL SCIENCE ONE and the Facebook Ad Library (formerly Ad Archive) since 2018.
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).
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.
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.
5. 'Deprecation' refers to the (scheduled) removal of nodes or edges, even if replacements were introduced simultaneously. In the latter case, we use the 'Replacement' label instead.
6. While we found some examples, 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.

References

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

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

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

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

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

Basole, R. C. (2018). On the evolution of service ecosystems: A study of the emerging API economy. In P. P. Maglio, C. A. Kieliszewski, J. C. Spohrer, K. Lyons, L. Patrício, & Y. Sawatani (Eds.), *Handbook of Service Science, Volume II* (pp. 479–495). Springer. https://doi.org/10.1007/978-3-319-98512-1_21

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

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

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

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

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-twitter-api

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, European Commission. https://platformobservatory.eu/app/uploads/2021/03/05Platformpower.pdf

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

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

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/

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

Dal Bianco, V., Myllärniemi, V., Komssi, M., & Raatikainen, M. (2014). The role of platform boundary resources in software ecosystems: A case study. *2014 IEEE/IFIP Conference on Software Architecture*, 11–20. https://doi.org/10.1109/WICSA.2014.41

van Dijck, J. (2020). Seeing the forest for the trees: Visualizing platformization and its governance: *New Media & Society*. https://doi.org/10.1177/1461444820940293

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

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. *Proceedings of the 12th ACM SIGSOFT Twelfth International Symposium on Foundations of Software Engineering*, 221–230. https://doi.org/10.1145/1029894.1029925

Eaton, B., Elaluf-Calderwood, S., Sorensen, 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

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

Freelon, D. (2018). Computational Research in the Post-API Age. *Political Communication*, 35(4), 665–668. https://doi.org/10.1080/10584609.2018.1477506

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

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

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

Fuller, M. (2008). *Software Studies: A Lexicon*. MIT Press.

Gawer, A. (2020). Digital platforms' boundaries: The interplay of firm scope, platform sides, and digital interfaces. *Long Range Planning*. https://doi.org/10.1016/j.lrp.2020.102045

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

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

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://computational-culture.net/regramming-the-platform/

Gillespie, T. (2018). Regulation of and by platforms. In J. Burgess, T. Poell, & A. Marwick (Eds.), *The SAGE Handbook of Social Media* (pp. 196–212). SAGE Publications.

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

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

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

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

Helmond, A., Nieborg, D. B., & van der Vlist, F. N. (2019). Facebook's evolution: Development of a platform-as-infrastructure. *Internet Histories*, 3(2), 123–146. https://doi.org/10.1080/24701475.2019.1593667

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/

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

Hogan, B. (2018). Social media giveth, social media taketh away: Facebook, friendships, and APIs. *International Journal of Communication*, 12(0), 20.

Isaac, 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

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/

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

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

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

Levy, S. (2014, April 30). Mark Zuckerberg on Facebook's future, from virtual reality to anonymity. *Wired*. https://www.wired.com/2014/04/zuckerberg-f8-interview/

Lomborg, S., & Bechmann, A. (2014). Using APIs for data collection on social media. *The Information Society*, 30(4), 256–265. https://doi.org/10.1080/01972243.2014.915276

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

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

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. https://doi.org/10.1016/j.dcm.2021.100505

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

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

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

Perriam, J., Birkbak, A., & Freeman, A. (2019). Digital methods in a post-API environment. *International Journal of Social Research Methodology*. https://doi.org/10.1080/13645579.2019.1682840

Plantin, J.-C., Lagoze, C., Edwards, P. N., & Sandvig, C. (2016). 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

Pridmore, J. (2016). A social API for that: Market devices and the stabilization of digital identities. In I. van der Ploeg & J. Pridmore (Eds.), *Digitizing Identities: Doing Identity in a Networked World*. Routledge.

Puschmann, C., & Ausserhofer, J. (2017). Social data APIs: Origin, types, issues. In M. T. Schäfer & K. van Es (Eds.), *The Datafied Society: Studying Culture Through Data* (pp. 147–154). Amsterdam University Press.

Pybus, J., & Coté, M. (2021). Did you give permission? Datafication in the mobile ecosystem. *Information, Communication & Society*, 0(0), 1–19. https://doi.org/10.1080/1369118X.2021.1877771

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

Rieder, B. (2013). Studying Facebook via data extraction: The Netvizz application. *Proceedings of the 5th Annual ACM Web Science Conference*, 346–355. https://doi.org/10.1145/2464464.2464475

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

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

Santos, W. (2019, October 24). Which are developers favorite APIs? *ProgrammableWeb*. https://www.programmableweb.com/news/which-are-developers-favorite-apis/research/2019/10/24

Schreieck, M., Hein, A., Wiesche, M., & Krcmar, H. (2018). The challenge of governing digital platform ecosystems. In C. Linnhoff-Popien, R. Schneider, & M. Zaddach (Eds.), *Digital Marketplaces*

Unleashed (pp. 527–538). Springer. https://doi.org/10.1007/978-3-662-49275-8_47

Skeggs, B., & Yuill, S. (2015). The methodology of a multi-model project examining how Facebook infrastructures social relations. *Information, Communication & Society*, 19(10), 1356–1372. https://doi.org/10.1080/1369118X.2015.1091026

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

Tiwana, A. (2014). *Platform Ecosystems: Aligning Architecture, Governance, and Strategy*. Morgan Kaufmann.

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

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

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

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

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

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

Zittrain, J. (2008). *The Future of the Internet—And How to Stop It*. Yale University Press.

Appendix

Table A-1. References to original and archived web sources.

In-text citation	Type	Reference
FB-2020	Business	(2020, August 11). Simplifying targeting categories. Available at: https://www.facebook.com/business/news/update-to-facebook-ads-targeting-categories/.
FB-2021*	Business	(n.d.). Facebook ads: Online advertising on Facebook. Available at: https://www.facebook.com/business/ads.
FD-2006a	Developers	(2006, October 10). Facebook Developers News. Available at: 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. Available at: https://web.archive.org/web/20061213220159/http://developers.facebook.com/background.php.
FD-2007	Developers	(2007, February 17). Extended permissions. Available at: https://web.archive.org/web/20070217011936/http://developers.facebook.com/news.php.
FD-2008	Developers	(2008, September 6). Extended permissions. Available at: https://web.archive.org/web/20080906150906/http://wiki.developers.facebook.com/index.php/Extended_permissions.
FD-2009	Developers	(2009, June 18). Application Verification Program. Available at: 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. Available at: https://developers.facebook.com/blog/post/417/.
FD-2010b	Developers	(2010b, April 23). Extended Permissions. Available at: https://web.archive.org/web/20100423170819/http://developers.facebook.com/docs/authentication/permissions.
FD-2010c	Developers	(2010c, May 8). Extended Permissions. Available at: https://web.archive.org/web/20100508225433/developers.facebook.com/docs/authentication/permissions.
FD-2010d	Developers	(2010d, August 31). Developer roadmap. Available at: https://web.archive.org/web/20100831180503/http://developers.facebook.com/roadmap.
FD-2010e	Developers	(2010e, December 5). Authentication. Available at: https://web.archive.org/web/20101205122001/http://developers.facebook.com/docs/authentication/.
FD-2010f	Developers	(2010f, December 12). Platform Upgrade Guide. Available at: https://web.archive.org/web/20101205155259/http://developers.facebook.com/docs/guides/upgrade.
FD-2011a	Developers	(2011a, August 31). A stable Platform. Available at: https://developers.facebook.com/blog/post/550/.
FD-2011b	Developers	(2011b, October 11). Breaking change policy. Available at: 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. Available at: 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. Available at: https://developers.facebook.com/blog/post/2014/10/30/graph-api-v2.2/.
FD-2014c	Developers	(2014c, May 21). Facebook Platform Changelog. Available at: https://web.archive.org/web/20140521164802/https://developers.facebook.com/docs/apps/changelog.
FD-2016	Developers	(2016, January 20). Targeting Specs. Available at: https://web.archive.org/web/20160120033705if_/https://developers.facebook.com/docs/marketing-api/targeting-specs/v2.4.
FD-2017a	Developers	Oakes Dunn, A. (2017a, September 20). Self-reported targeting. Available at: https://developers.facebook.com/ads/blog/post/2017/09/20/self-reporting-targeting/.
FD-2017b	Developers	Bala, D. (2017b, December 4). Targeting exclusions. Available at: https://developers.facebook.com/ads/blog/post/v2/2017/12/04/targeting-exclusions-blog-post/.
FD-2017c	Developers	(2017c, February 25). Targeting Specs. Available at: 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'. Available at: 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. Available at: 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. Available at: 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. Available at: 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. Available at: https://developers.facebook.com/blog/post/2018/12/10/verification-for-individual-developers/.
FD-2018f	Developers	(2018f, April 4). April 4, 2018. Available at: https://developers.facebook.com/docs/graph-api/changelog/non-versioned-changes/apr-4-2018#login-4-4.
FD-2018g	Developers	(2018g, May 1). Version 3.0: App Review. Available at: https://developers.facebook.com/docs/graph-api/changelog/version3.0#new-app-review.
FD-2018h	Developers	(2018h, March 9). Targeting. Available at: 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. Available at: https://developers.facebook.com/docs/graph-api/changelog/version6.0.
FD-2021a*	Developers	(n.d.). Developer Policies. Available at: https://developers.facebook.com/devpolicy/.
FD-2021b*	Developers	(n.d.). Release. Available at: https://developers.facebook.com/docs/development/release.
FD-2021c*	Developers	(n.d.). Changelog. Available at: https://developers.facebook.com/docs/facebook-login/changelog.
FD-2021d*	Developers	(n.d.). Overview. Available at: https://developers.facebook.com/docs/facebook-login/overview.
FD-2021e*	Developers	(n.d.). Permissions with Facebook Login. Available at: https://developers.facebook.com/docs/facebook-login/permissions/overview.
FD-2021f*	Developers	(n.d.). App Review. Available at: https://developers.facebook.com/docs/facebook-login/review.
FD-2021g*	Developers	(n.d.). Changelog. Available at: https://developers.facebook.com/docs/graph-api/changelog/.
FD-2021h*	Developers	(n.d.). Overview. Available at: https://developers.facebook.com/docs/graph-api/overview/.
FD-2021i*	Developers	(n.d.). Graph API User. Available at: https://developers.facebook.com/docs/graph-api/reference/user/.
FD-2021j*	Developers	(n.d.). Audiences. Available at: https://developers.facebook.com/docs/marketing-api/audiences.
FD-2021k*	Developers	(n.d.). Special Ad Category. Available at: https://developers.facebook.com/docs/marketing-api/audiences/special-ad-category/.
FD-2021l*	Developers	(n.d.). Permissions Reference. Available at: https://developers.facebook.com/docs/permissions/reference.
FD-2021m*	Developers	(n.d.). Permissions Reference: Instagram Permissions. Available at: https://developers.facebook.com/docs/permissions/reference#instagram_permissions.
FD-2021n*	Developers	(n.d.). Permissions Reference: Facebook Login Permissions. Available at: https://developers.facebook.com/docs/permissions/reference#login_permissions.
FD-2021o*	Developers	(n.d.). Facebook Platform Terms. Available at: https://developers.facebook.com/terms/.
FL-2019a	Leaks	Campbell, D. (2019, November 6). Facebook leaks. Available at: https://www.duncancampbell.org/facebookleaks.
FL-2019b	Leaks	Campbell, D. (2019, November 6). Facebook leaks (pp. 2–70). Available at: 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). Available at: 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). Available at: https://dataviz.nbcnews.com/projects/20191104-facebook-leaked-documents/assets/facebook-sealed-exhibits.pdf.
FL-2019e	Leaks	Campbell, D. (2019, November 6). Facebook leaks: Exhibit 78 (p. 802). Available at: https://dataviz.nbcnews.com/projects/20191104-facebook-leaked-documents/assets/facebook-sealed-exhibits.pdf.
FL-2019f	Leaks	Campbell, D. (2019, November 6). Facebook leaks: Exhibit 104 (pp. 1356–1393). Available at: 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). Available at: 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). Available at: 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. Available at: 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. Available at: https://about.fb.com/news/2019/11/introducing-our-new-company-brand/.
FP-2021a*	Policies	(n.d.). Advertising Policies. Available at: https://www.facebook.com/policies/ads/.
FP-2021b*	Policies	(n.d.). Advertising Policies: 3. Discriminatory practices. Available at: https://www.facebook.com/policies/ads/prohibited_content/discriminatory_practices.
IE-2014	Engineering	(2014, June 25). Migrating from AWS to FB. Available at: https://instagram-engineering.com/migrating-from-aws-to-fb-86b16f6766e2.

* Date accessed 23 March 2021.

Author biographies

Fernando van der Vlist is a PhD candidate at Utrecht University and a research associate with the Collaborative Research Centre ‘Media of Cooperation’ at the University of Siegen. His research interests include social media and platform studies, digital methods, app studies and mobile media, and software studies.
ORCID: 0000-0003-1401-0325
Twitter: @fvandervlist

Anne Helmond is an assistant professor of New Media and Digital Culture at the University of Amsterdam. Her research interests include software studies, platform studies, app studies, digital methods, and web history.
ORCID: 0000-0003-4327-4012
Twitter: @silvertje

Marcus Burkhardt is a lecturer at the chair for Digital Media and Methods, University of Siegen. His research focuses on the history and theory of digital media, especially the logi(sti)cs of database technologies, big data, and algorithmic environments as well as on media of knowledge production and dissemination, media philosophy and media theory.
ORCID: 0000-0003-0382-3128
Twitter: @bumatic

Tatjana Seitz 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.
Twitter: @taz_seitz