Super SDKs: Tracking personal data and platform monopolies in the mobile

Jennifer Pybus, York University Toronto

Mark Coté, Kings College London

April 2024
Super SDKs: Tracking personal data and platform monopolies in the mobile

Jennifer Pybus1 and Mark Coté2

Abstract

In this article we address the question 'what is tracking in the mobile ecosystem' through a comprehensive overview of the Software Development Kit (SDK). Our research reveals a complex infrastructural role for these technical objects connecting end-user data with app developers, third parties and dominant advertising platforms like Google and Facebook. We present an innovative theoretical framework which we call a data monadology to foreground this interrelationship, predicated on an economic model that exchanges personal data for the infrastructural services used to build applications. Our main contribution is an SDK taxonomy, which renders them more transparent and observable. We categorise SDK services into three main categories: (i) Programmatic AdTech for monetisation; (ii) App Development, for building, maintaining and offering additional artificial intelligence features and (iii) App Extensions which more visibly embed third parties into apps like maps, wallets or other payment services. A major finding of our analysis is the special category of the Super SDK, reserved for platforms like Google and Facebook. Not only do they offer a vast array of services across all three categories, making them indispensable to developers, they are super conduits for personal data and the primary technical means for the expansion of platform monopolisation across the mobile ecosystem.

Keywords

Software development kits, datafication, platforms, privacy, apps, surveillance

Introduction

In November 2022, Google paid a 391.5 million USD fine after a coalition of 40 states proved the platform had been deliberately misleading users for years, tricking them into believing they had turned off their location data when in fact it was being collected all along. Under this privacy settlement, Google promised to make the disclosure of location tracking clearer (Kang, 2022). While this seems an important first step, how it will work remains to be seen, especially given that users do not have a singular but multiple data points of contact with platforms. Each time we use Google’s search algorithm, its maps, or its applications (apps), the tech giant gathers a wide array of our personal data. But Google is also inside almost every app in the mobile ecosystem, providing tools and services to help developers make and monetise apps. Google’s SDKs (Software Development Kits) are found in more than 93% of mobile apps (Statista, 2023), creating a pervasive network for gathering and cross referencing not only location, but all kinds of behavioural and other personally identifiable data.

This article is guided by a deceptively simple question: what is tracking in the mobile and how does it work in our apps? We address aspects of this by examining SDKs, technical objects which provide development tools, Application Programming Interfaces (APIs) and advertising services to app developers. Often, SDKs are referred to as ‘trackers’. But this term does not capture the complexity of services that they offer, neither how widely their services are embedded by app developers, nor the sheer range of ways in which they may use the data they extract. To bring more visibility to this complex and opaque infrastructure, we present SDKs within a novel theoretical framework: a data monadology. Here, paraphrasing Latour et al. (2012), we ask: what does the small of the SDK reveal about the whole of the expansionary logic of data-powered capitalism in mobile applications? Studying how SDKs are situated, opens a way to engage an infrastructure that has been calibrated to...
maximise a distributed service-for-data economy (Blanke and Pybus, 2020) within our apps. Thus, to better answer this question, we provide an SDK service taxonomy that we have developed to facilitate a new line of research on mobile infrastructures and platforms. We use the taxonomy to analyse a single application to illustrate the expansive integration of SDks, including Google’s primary mobile assets: Firebase and Admob. We highlight how SDks provide a constellation of services in any given app, with the potential to capture and process user data in myriad ways. We also put forward a unique category, the Super SDK, distinguished by the significant range of services they provide in every category of our taxonomy. These appear to belong uniquely to large platforms like Facebook and Google. Finally, we identify a series of related regulatory privacy challenges that stem from their lack of visibility to end-users and the opacity challenges they present to developers and policy makers.

What are SDks?

Defining SDks is both a simple and a complex task. On the one hand, these can be understood as straightforward trackers, fundamental to surveillance capitalism (Zuboff, 2019). But for developers, SDks are practical technical objects that offer a complexity of services. They represent the basic building blocks for app creation – modular connectors that come preloaded with ready-made tools, code, libraries and tutorials from either platforms and/or third parties that are integral to the software supply chain (Gontovnikas, 2020). Additionally, they usually contain APIs, making it easy for developers to connect and draw directly from these external services. It is important to emphasise that SDks are not APIs, but rather encase them, allowing developers to pick and choose what they need to create and/or monetise their apps.

Critical scholarship on the political economy of SDks has focused on the ways in which they facilitate invisible partnerships that enable data access and distribution – particularly for platforms – by further integrating a constellation of actors that make-up the digital advertising ecosystem (Blanke and Pybus, 2020; van der Vlist and Helmond, 2021). However, SDks are not just used for monetisation, they are also required to build and maintain apps – from coding languages and building environments; to cloud storage and support; to basic crash support and security – which remains overlooked and understudied. Additionally, what is not well understood are the ways in which the various SDk services that a developer chooses to integrate, be these for monetisation and/or for building and maintaining apps, create new pipelines for third parties or platform to gather personal data. In practical terms, this infrastructure complicates the ability of developers to prevent or even know what is being accessed from a user’s device for two reasons: (1) SDks are black boxed, so developers must trust, instead of know if they are compliant with data protection legislation and (2) because this infrastructure is embedded, when developers make requests for user data, SDks are given access too
t (Tahaei et al., 2022). From this perspective, not only do they largely evade privacy regulation, but developers are made responsible for any data that they may extract. Perhaps this infrastructure has been overlooked because its original function was primarily to connect apps to app stores (Morris and Elkins, 2015). However, SDks have come a long way, and offer increasingly sophisticated services such as advanced artificial intelligence (AI) models for facial recognition and natural language processing, as well as new techniques to track users more comprehensively across all devices, including via Internet of Things (IoT). Our article thus addresses the question of what is tracking in the app ecosystem by introducing a service taxonomy that we developed to render the SDks more transparent and observable (Rieder and Hofmann, 2020) for further research and enquiry on the myriad ways personal data is captured and rendered actionable in the mobile ecosystem.

The small of the SDK is bigger than the whole of the platform

We situate our service taxonomy in the conceptual frame of a data monadology. This novel orientation positions this small piece of infrastructure as a unique vantage point for seeing the whole of the tracking ecosystem’s expansionary logic. This approach builds on the atomistic natural philosophy of Epicurus and Lucretius, the computational calculus of Leibniz’s infinitesimal, and Tarde’s (2010) monad. Recently, this can be seen through Latour (2002), Terranova (2012) and Lazzarato (2004) for whom the understanding of social connections and/or the social production of value comes not from knowing the whole but from the elements that bind, connect and possess one another. Inferences, similarly, emerge through the amalgamation of attributes composed of preferences and opinions, enabling behavioural prediction that facilitate targeted advertising. SDks provide the connective infrastructure to platforms through which our personal data flows and is activated in a service economy made up of platforms and third parties or complementors (Blanke and Pybus, 2020; Van Dijck et al., 2018).

A data monadology conceptualises connectivity via interoperability. From this perspective, increasing access to user data, means increasing access to attributes, thereby increasing opportunities to produce a multiplicity of actionable profiles for a range of agents across our apps, platforms and devices. A data monadology frames the algorithmic processing of users as a composition of attributes that can be (re)combined in and of themselves
and in relation to the attributes of other users at scale to maximise surplus value and exchange. For example, take the Google Firebase SDK present in more than 9 out of 10 apps. It enacts broad and deep networked effects, linking users, apps, operating systems, ad networks, cloud computing, AI and Machine Learning (ML) capacities, and other services in an extensive and friction-free manner. Furthermore, it connects, actions and capitalises on this capacity to (re)combine the attributes of its users, who are integrated through multiple points of contact. The logic that Firebase encapsulates benefits from the non-rivalrous qualities of data (Rieder, 2022), wherein personalisation is neither one-to-one nor whole-to-whole, but rather attribute-to-attribute with infinite (re)compositions. The monad draws our attention to the attributional relationship between inference and prediction – combining past behaviours of the many, to predict the future behavioural profiles of the few – maximising the plurality and distribution of attributes that can be extracted and made actionable.

A data monadology allows us to imagine how any actor interested in our personal data might see us: not as a whole, not as a unique individual, but rather as a composition of nonrivalrous attributes that can be (re)used and (re)combined depending on the profile required for advertising, news feeds, and so on. To start at the level of the SDK, helps to visualise this expansive capitalistic logic that creates infinite number of opportunities to maximise access to end-user data – no matter how small or innocuous. The SDK implements sociotechnical networks, ascribing standards, proprietary protocols, and code for maximising the capture, generation and aggregation of value from data, and from which inferences and predictions render individuals knowable and actionable. We therefore see a conceptual clarity in the ‘monad’ that Latour takes from Tarde to develop his actor-network theory (Latour, 2002). However, instead of linking and understanding the networks that underscore the social, like Latour, we apply this logic to the nonrivalous SDK service-for-data economy, captured in the taxonomy.

The provision of services-for-data is dependent on the infrastructure provided by SDKs; thus we can categorise this economic exchange as a Service-as-an-Infrastructure (SaI) model. In other words, SDKs create opportunities for personal data attributes to be continuously transformed and reused, scaling up lucrative inferences and predictions for multiple actors across the data economy. SaI foregrounds how integral SDKs are to both app design and the expansion of end-user tracking, illustrating both the role of platforms (Plantin et al., 2018) and other third parties which participate in the mobile’s service-for-data economy. We have deliberately played on the economic cloud service pay-as-you-go model: ‘Infrastructure-as-a-Service’, which offers different computational, storage and networking resources on demand, such as those provided by Microsoft or Amazon. We do so to emphasise how this infrastructural service provision enacts and expands datafication in the mobile ecosystem.

### A short history of tracking: From web cookies to mobile SDKs

To understand what is distinct about SDKs and their relationship to platforms, let us consider how they originated as ‘trackers’. Part of this history begins in the 1990s, when pressure was mounting on platforms to find ways to leverage their assets and guarantee return on investments (West, 2019). Part of this problem was that the web was never built to remember, only to connect. In 1994, Netscape framed this as the ‘shopping basket problem’ (Schwartz, 2001), that is, how to remember user activity once they navigate away from your website? Advertisers wanted assurances. Who was seeing their ads? Where? Did they work? What could be used to demonstrate return on their investments? Through the mid-1990s there were no metrics, and thereby no infrastructure to attribute ad spending with conversion. Hence the critical importance of the first cookie, a small text file that could attach itself onto any given browser and recall what the web could not. Think of the cookie as a prosthetic memory device: a bridge between a user’s browser and the server with an important accolade – the ability to tag, store and pass on user data. Cookies thus transformed advertising and empowered digital networks to disrupt print and broadcast media as a new infrastructure for content provision (Turow, 2011), a history richly documented (McStay, 2017; Mellet and Beauvisage, 2019; West, 2019; Zuboff, 2019). In conjunction, was the rise of platforms (Snircek, 2017; Van Dijck et al., 2018) which were transforming the production, distribution and circulation of content (Nieborg and Poell, 2018; Poell et al., 2021), expanding their reach through affiliations with third-party partnerships (Helmond et al., 2019), enabled by infrastructures built to support various intermediaries (Braun, 2013) to make data ‘platform-ready’ (Helmond, 2015; Poell et al., 2019). Platforms made important use of cookies, giving rise to an explosion of actors who make up the digital advertising or programmatic AdTech ecosystem, designed to get the right ad, to the right person, at the right time, on the right screen. This invasive infrastructure brought forward a range of privacy and governance challenges (Geradin and Katsifis, 2019; Gorwa, 2019; Veale and Borgesius, 2022) resulting from how increasing value is derived from the accumulation and processing of personal data.

What is less developed in these histories of platform-based digital capitalism (Snircek, 2017; West, 2019; Zuboff, 2019) is the unique materiality of the mobile ecosystem. Within app studies, these infrastructures are well documented as being both situated and embedded (Dieter et al., 2019). We have argued that these infrastructures
have expanded platform monopolisation (Blanke and Pybus, 2020), quietly intensifying the cycle of data capture, inference production and commodification of predictions at scale. The closed environment of the mobile is a feature not a bug: everything is embedded, allowing any given app to cultivate a much more intimate relationship with end-users. The SDK was important because of the ways in which it interconnects apps with third parties and platforms in a distributed network. Unlike cookies, SDKs need not help apps remember but rather help app developers connect in exchange for user data. They are both integral to and integrated inside every app. We put them forward as a crucial agent of datafication; critical infrastructure facilitating the distribution of user data to a range of actors who in turn produce their own respective value and insights in what Flensburg and Lai (2022) call the ‘appscape’.

The rise of SDKs occurred soon after the 2007 launch of the iPhone, initially preloaded with native applications: clock, stocks, calendar, etc. Apple wanted third parties to create ‘web apps’ for their Internet browser Safari, which then could be accessed by the iPhone. The first iOS SDK was designed to enable phone-web interoperability. Developers however, resisted, wanting apps to be housed directly on the phone, not on the web. Enter the first true mobile applications: developers conspired to ‘jailbreak’ the iOS SDK and directly install Apple’s platform specific tools and software inside their own apps (Morris and Elkins, 2015). Apple quickly recognised an opportunity and changed course. The iOS SDK was thus transformed into an installable, interoperable package that ultimately led to the launch of Apple’s App Store in March 2008. Subsequently, it produced an initial wave of more than 500 apps from third-party developers. Later that year, Google followed with its own SDK and opened the Google Play store. Today, Apple has more than 1.65 million apps, while Google has more than 3.5 million (Ceci, 2022). What unites the web browser and mobile app is the prominence of platforms, using both cookies and SDKs as infrastructures of capitalisation and monetisation of user data. While cookies enabled the initial establishment and growth of platforms within the browser, SDKs have extended their dominance across mobiles.

**Privacy problems with SDKs**

Studying the backend of mobile applications is challenging, however, there have been important contributions by app studies scholars who have opened up ways of studying these unique infrastructures. For example, we see approaches for technical backend walk-throughs (Duguay and Gold-Apel, 2023); studies on mapping and understanding app permission and third-party libraries (Dieter et al., 2021; Flensburg and Lai, 2022; Helles et al., 2020; Pybus and Coté, 2021; Weltevrede and Jansen, 2019); on digital objects like ‘manifest files’ which govern an app’s access to the personal data stored on an end-user’s mobile device (Dieter et al., 2019; Pybus and Coté, 2021) or ways of opening up more sustained infrastructure literacy approaches (Gray et al., 2018). And yet, SDKs remain largely opaque, with a paucity of auditing tools to fully account for the ways in which they access personal data from inside of our apps. Subsequently, users are left at undue risk, with little to no recourse to identify the actors who are routinely accessing their data, despite an expectation to the contrary set by current data privacy legislation.

This privacy problem is compounded by the growing evidence of ‘widespread infringements’ of both GDPR and other data protection laws (Kollnig and Binns, 2022). Violations are evidenced in the ways in which apps do not adequately comply with the Children’s Online Privacy Protection Agreement (Reyes et al., 2017); fail to provide adequate privacy policies (Kollnig, 2021); fail to comply with their own privacy agreements (Story et al., 2019); are inconsistent in the ways in which they disclose how parties share and transmit secure data (Okayomon et al., 2019); share more data than necessary with third parties and send user data to countries with less adequate levels of data protection (Kollnig et al., 2021). Furthermore, we see no evidence of privacy legislation requiring app developers to disclose which third parties are silently present, processing user data, raising the question of why the privacy policies of apps do not specifically name these third parties?

In the cybersecurity literature Zimmelck et al. (2016) analysed almost 18,000 apps, noting as many as 17% could be sharing data such as location with third parties without disclosing this information to users. Other studies, such as one done on mental health apps show 41% of apps examined did not acknowledge how personal user data would be collected, retained, or shared with third parties despite this being a data protection policy requirement (Parker et al., 2019). Similarly, Mehrnezhad et al. (2022) note that apps that serve women’s health needs, also known as FemTech, aggregate sensitive health data, often sharing this with advertisers and data management platforms without adequate consent. The reality is, if users are provided with a privacy policy about their apps, and if these refer to third parties, there is still no meaningful visibility afforded to the actions of third-party SDKs. In short, language commonly used, like in Strava’s privacy agreement: ‘Information collected by these third parties is subject to their terms and conditions. Strava is not responsible for the terms or policies of third parties (Strava, n.d., our emphasis)’, does not name or meaningfully account for the personally identifiable data their SDKs access.

Summarising, there are several privacy challenges for mobile applications which go beyond the scope of this article. Given that data protection legislation describes personal data as a ‘special category’ and thereby a protected
attribute, there needs to be more scrutiny of the ways in which SDKs are facilitating the building of profiles, audience segments, targeting techniques or inferences from combining data with other third-party data sources. We see SDKs as occupying a grey zone. They can inherit the full range of user data for which the developer has asked for their permissions, largely evading effective legislation or scrutiny. Their plug-and-play nature means developers do not need to write code to access their services, which in turn can lead to a higher prevalence of privacy violations, even when there is an effort made to adhere to privacy regulations (Tahaei et al., 2022, 2023; van der Linden et al., 2020). This raises ethical questions around not just the degree to which developers can be held responsible for data privacy but of broader asymmetrical relations that platform power enacts therein. We therefore see a privacy gap in the interactions between our apps and SDKs, wherein the personal data they gather are mostly invisible to users, developers, regulators and auditors. Our intervention is to develop a taxonomy of services offered by SDKs, providing some clarity to the data practices of this unique intermediary. We anticipate this will provide greater interpretability for the means by which datafication is enacted between SDK services and mobile applications.

The SDK taxonomy

The SDK taxonomy (Figure 1) we developed categorises the services offered by platforms and other third parties to developers. We began by performing a comprehensive literature review on the scholarly work that have provided ways to distinguish and classify the ‘trackers’ in the mobile ecosystem. In a comprehensive study of over one million Android apps, third parties are referred to as trackers that provide mainly ‘analytic’ and/or ‘advertising’ support (Binns et al., 2018). Flensburg and Lai (2022) additionally foreground how SDKs provide ‘social media plug-ins’. Feal et al. (2020) provides one of the most comprehensive working lists of services, noting that SDKs not only facilitate advertising and monetisation but also offer critical infrastructural support to developers. Secondly, we consulted advertising and marketing literature to understand industry nomenclature. Key sources included blogs like Mighty Signal (now AirTime), AppFigures and Statista which curate and provide SDK ‘leader’ lists based on specific services that developers might require. We then mapped the services that were found in 10 of the most commonly used SDKs (see Table 1) and triangulated the results by drafting an initial set of categories for the taxonomy. Thirdly, we applied our draft taxonomy to a series of SDKs and made iterative adjustments. For example, we initially had analytics as its own category, but quickly realised that almost every service we looked at provided an array of analytics, making this category too vague and encompassing. And finally, we mapped our taxonomy onto a single app, to test what the small of the SDK could reveal about the whole of the tracking ecosystem.

For our case study, we selected the clothing app ASOS for three reasons: (i) this popular European app was chosen for inspection by many students in workshops we have led; (ii) it has a strong distribution of the different services to illustrate the value of our taxonomy and (iii) it has 18 SDKs, the industry average. Next, we inspected two of Google’s key assets found in this app – Firebase and Admob – given their prevalence within the app ecosystem. In so doing, we aimed to demonstrate how a taxonomy can enhance qualitative research within the mobile ecosystem. It should finally be noted that this is a living taxonomy,
which we hope will be developed further through new research in this field.

Our taxonomy categorises SDK services into three main clusters: (i) Programmatic AdTech which is entirely for monetisation; (ii) App Development, which developers rely on to build, maintain and enhance the features of their applications and lastly (iii) App Extensions which are embed specific features into apps like a social logins, maps or payment services. Our analysis reveals that while many of these companies are quite specific and targeted such as Chartboost, which specialises in the creation and mediation of playable video ads, others have much more extensive offerings. We therefore propose one more all-encompassing service category: the Super SDK, reserved for platforms like Facebook and Google which provide multiple services in all three categories, rendering them indispensable when it comes to the app economy. To be clear, this taxonomy is valuable not just for clarifying the different services provided but to deepen our understanding of how tracking works and to more systematically address the governance challenges posed by SDKs. Below is a more detailed overview of the categories we have created for the taxonomy.

### SDK Service categories

**Programmatic AdTech**

Central to the monetisation of apps is programmatic advertising, a highly complex and specialised digital infrastructure, composed of multiple actors accumulating user data for targeted advertising (McStay, 2017). This is a highly scalable process that ultimately leverages inferences derived from analytic insights about the aggregated behaviour of users, which in turn transformed into predictions that can be bought, sold and metricised based on effectiveness. To produce this future oriented marketplace (Zuboff, 2019), several actors have emerged that enable the accumulation, combination and (re)use of user data, which for us lies at the heart of datafication. Our taxonomy distills these down to three dominant types of services that procure and action data and analytics in different ways – Attribution, Engagement and Advertising.

**Attribution.** These services attribute which ads were seen, by which people, on which media channel, via which campaign. To attribute is, by definition, to track. These SDKs facilitate the entirety of what advertisers commonly refer to as the ‘conversion funnel’ or the ‘customer journey’. This begins the moment of seeing an ad or a product to the moment of its purchase or installation. The programmatic advertising industry depends on attribution to seamlessly follow users across all their devices, ad networks and platforms. These SDKs, like Apps Flyer, Branch, and Adjust, are also used to detect fraudulent accounts and/or bots to guarantee and further legitimise their services. For existing privacy legislation, attribution SDKs pose a significant challenge. Its *raison d’être* is the development of more agile and pervasive digital surveillance. Over the past 3 years, we have seen the rise in new metrics, such as Apple’s PCM (privacy click measurement), in addition to Google’s proposed privacy preserving ‘attribution reporting API’. Both purport to hide personally identifiable information about users from third parties by processing anonymously attributed data on devices. However, regardless of a person’s anonymity, attribution cannot work without invasive and intimate tracking and reporting about all our behaviour to legitimise ROI for ad spending.

**Engagement.** These are services intimately tied to profile building. Using analytics called ‘app events’, they track everything we do, in real-time, in our apps. This data is used to generate psychographics, grouping us by interests,
habits and other kinds of personal preferences. To enhance these profiles, engagement SDKs like Urban Airship, Clever Tap, Adobe or Braze gather user data from multiple points of contact, in different applications, which are then (re)combined in their own respective centralised database – an area that needs further research. Their metrics help optimise in-app targeted advertising by providing insight to developers for audience investment, and for adapting content presentation. Engagement software provides a range of A/B testing services for ads, products, UX design, campaigns, or push notifications for personalisation, and optimisation. Finally, these SDKs are designed to track and nudge users across platforms and devices by accessing personal user data, namely email addresses, phone numbers, location and/or various advertising IDs to maximise engagement with their notifications, reminders and dynamic links designed to direct the user to back the app regardless of what device they are on.

Advertising. These services produce and/or integrate ads in addition to facilitating in-app purchases. Mediation is one of the main services offered, which gives developers more flexibility and control about who can purchase their ‘inventory’ or ad space in the app. In practical terms, this means getting paid more for an ad with a higher CPM (cost per mille or one thousand impressions). SDKs like Admob, Facebook Ads, ApplLovin or Chartboost are useful because they provide single-point ad network integration – mediators between publishers and advertisers – which bid for an app’s available inventory (designated space for an ad to appear), making it easier for developers to maximise advertising revenues. These SDKs also specialise in the creation and integration of ads, such as interstitial or reward ads that are used to enhance user engagement, especially for gaming apps. Finally, many offer ‘brand safety’ tools that can block ‘inappropriate’ ads or prevent their placement with inappropriate content.

App development

Building apps requires the integration of APIs, cloud services, libraries and code offered by SDKs. To ensure these complex services are used correctly, they typically come with libraries, tutorials and troubleshooting guidelines. While personal data is not directly gathered by these SDKs, the fact that they are integrated into so many different applications means that metadata can still be used to track people via the app IDs they gather (Monogios et al., 2020; Razaghpahang et al., 2018). We have broken this cluster down into app creation, cloud access and database support, and AI and ML services.

App creation. This broad category includes the constellation of actors that create and support apps. Here we find SDKs that provide cross-app development environments like Unity, Flutter (Google), React Native (developed and maintained by Facebook), Xamarin or Amazon’s AWS amplify. These are used to build, test and deploy the creation of apps for multiple operating systems that have unique code requirements such as Swift for iOS apps and Java or Kotlin for Android apps. In this sense, they can facilitate interoperable app creation and design across platforms. These SDKs also offer a range of (specialised) code such as Airbnb’s font animation library called Lottie, located in the open-source React Native Directory (React Native Directory, n.d.). They also provide specialised features like Firebases’ heart rate monitor or key components to support programmatic AdTech like the push notifications or in-app messaging services offered by Firebase, Facebook, OneSignal or UrbanAirship. Other built in AdTech features might include the integration of deep links that are offered by SDKs like Branch, Adjust or Adobe to directly send users to an app, instead of a website or a store. Finally, developers often embed these SDKs for their crash reporting services (checking for bugs and problems in their app code), like those offered by Crashlytics in Firebase, Flurry or New Relic.

Cloud access and database support. These offer apps intensive computation on demand. Their services include seamless database integration, the merger of data storage and data transfer, dynamic scalability, cloud hosting for profile building, real-time synchronisation, emulator/infrastructure for A/B testing and experimentation, all of which can be operating system agnostic. For example, Firebase recently released Cloud Firestore, which syncs and processes enormous amounts of data between devices both on-line and off-line. Similarly, Amazon’s AWS DynamoDB promises ‘more than trillions of requests per day and millions of requests per second’ (Samaranayake, 2021). Such services facilitate advanced applications and raise geopolitical concerns around the location of cloud servers and where and by whom user data is processed.

AI and ML services. These SDKs provide advanced ML capabilities to developers. Their services include access to machine vision for facial recognition, object recognition, landmark recognition, image labelling or barcode detection; natural language processing (NLP) for labelling, textual analysis, text recognition, language identification and translation; and sound analysis frameworks for analysing audio, sentiment classification and sound differentiation like birds singing or traffic (Bansal, 2022). We also expect that ChatGPT will eventually have its own SDK. At the time of writing this, open-source SDKs like Chat-GPT Wrapper exist which could in theory be used to integrate this feature into applications (Zhang, 2023) like SnapChat has done.
App extensions

These primarily include third-party services which are visible to users. Unlike Programmatic AdTech and App Development tools, App Extensions are clearly signposted and visible to end-users, removing any ambiguity about their existence. Notably, the difference between these SDKs and APIs in particular appear almost negligible. However, because they are integrated via the encased SDK infrastructure and often provide additional features, tools and guidance, they stand as their own category, based on the ways in which developers must embed their services. We have them in three sub-categories: authentication, social media-plugs, and other services, most notably of which are payment services.

Authentication. These SDK services offer security benefits to developers: protecting logins against unauthorised access, and user data. Login authentication normally entails emails and passwords; phone numbers; biometrics (face and/or fingerprints); or, federated identity providers, linking the user’s account to another identity account such as Facebook, Twitter, Google, or GitHub. These services enable access to personally identifiable data by linking digital identity across multiple management systems. Authentication using the Facebook SDK, for example, gives access to a user’s social graph, and allows requests of over 130 different permissions, including a user’s name, their profile photo, their birthdays, their friends, gender, hometown and so forth.4 This authentication data in and of itself is a rich resource for profile building and inference generation. Furthermore, biometric data are not entirely subject to either industry standards or regulations regarding access from apps and/or devices (Adra, 2018).

Social media plugs-ins. These are services that can be incorporated by the developer that make it easier for the user to post, share or connect with external platforms. In the past, these would easily allow platforms to gather data about users, although this has been limited somewhat in Europe. In July 2019, the Court of Justice of the European Union decided that Facebook would be liable for personal data exchanged via liking or sharing of content if users had not logged into their account or provided their consent (Lomas, 2019). We believe these are in the midst of being phased out and integrated into Super SDK services.

Other services. This is a broad category that accounts for a range of tools that apps embed that are signposted, such as maps or financial tools which may ask users to agree to their specific privacy agreements before activating their features for the first time. Some of these services include payment wallets such as Google Pay, AliPay or ApplePay; on-line payment portals such as Payment Tree and Forter; or access and integration with fintech companies who offer buy-now-pay-later options (BNPL) such as Klarna, AfterPay or PayPal.

The rise of the super SDK

The final category that emerged from our empirical analysis is what we call the Super SDK, namely for our study Google and Facebook.5 In previous research, we observed multiple discrete SDKs belonging to each platform (Blanke and Pybus, 2020). Now, however, Google Ads, Google Analytics, Double Click and Crashlytics have been streamlined into the Google Firebase SDK. What then makes them these SDKs ‘Super’ is their full interoperability across all of Google’s assets, comprehensive service coverage across all three taxonomic clusters, and finally the dependencies they create for developers who are reliant on the extensive services that they offer. Super SDKs thus function as primary hubs, conscribing connectivity. Furthermore, they supercharge the role of platform advertising giants by increasing, standardising and automating their access to vast amounts of intimate mobile data by offering the monetisation services on which developers have become dependent (Harlan, 2021). The capacity for these platform SDKs to generate value should not be underestimated. We can expand upon this by applying the perspective of data monadology.

If we undertake a monadological reading of Super SDKs, we can link the effectiveness and self-sufficiency of a monad’s activity to how Leibniz theorised its perfection. For Leibniz, ‘the more perfect a thing is, the more it acts and the less it is acted upon’ (Strickland, 2014). Super SDKs express their ‘perfection’ through the overwhelming degree to which they act upon app development. The increased number of services they offer and hence dependencies they afford, means they disproportionately impose standards and regulation of user data flows within apps. If we understand that value is not pre-formed in the ‘whole’ of the individual but emerges from the constellation of infinitesimal attributes generated across users, then Super SDKs – the monological agents par excellence – are set to maximise and action these profitable connections. Their technical affordances are precisely what platforms like Facebook or Google leverage for their digital advertising empires, putting our attributes into countless combinations with those belonging to every other user in their respective databases. In short: Super SDKs cohere data monads into profitable new ‘wholes’, optimally situated to capitalise, with all the assets and affordances that their platform power provides.

Applying the taxonomy: Looking inside the ASOS app

To illuminate the expansive ways in which we are connected to the data gathering practices of SDKs, we have
applied our taxonomy to ASOS. One might consider this close examination as a variation of the walk-through method (Duguay and Gold-Apel, 2023; Light et al., 2018). We, however, are opening the app to a new level of data granularity by inspecting embedded SDKs. Figure 2 demonstrates that ASOS has enlisted 18 third parties. First, we note the Super SDKs: Firebase, Android (both Google), and Facebook. Each of these SDKs are found in all three service clusters of the taxonomy. We also see the prominence of Urban Airship, another popular engagement SDK, specialising in personalised push notifications via the app, email and/or WhatsApp, and requests access to email addresses, phone numbers and browsing histories. Second, we observe 10 app development SDKs, with both Firebase and Facebook playing very prominent roles across all three categories in the clusters. Third, we see the presence of several financial services for in-app payments. Fourth, we observe another eight SDKs providing programmatic AdTech support, including the Chinese company Huawei, and three of Google’s key assets: Firebase, Admob and Android.

Our next step was to consult the ‘App Privacy’ report in the iOS app store and the ‘Data Safety Report’ in the Google Play Store to learn more about ASOS’s privacy agreement. Both note that the app uses search history and ‘identifiers’ to track users, with the following data gathered from each user profile: purchases, location (approximate and precise), address, phone number, email address, search history, usage data, financial info, identifiers, diagnostics, and other data for marketing purposes. A deeper look at both its privacy policies reveals that ASOS shares user data with ‘companies that do things to get your purchases to you’, including payment providers, brand providers, marketing agencies, affiliates used to reach customers and credit agencies. In neither the iOS nor in the Google Play Store does ASOS’s privacy policy give any indication of who their affiliates are, the number of SDKs they are using, that one SDK may be sending user data to China, nor is there any clarity about the kinds of data the affiliates of these SDKs can gather or aggregate with their own third-party affiliates. This last point raises yet another question: how do we label the third parties that SDKs share end-user data with? Should these known unknowns be called: the third parties of the third party? More research is needed to open up this important blind spot and consider how this governance challenge might be more clearly expressed within current data protection policies and regulations. Finally, coming back to ASOS, we note a disproportionate number of services being provided by Google and Facebook across the taxonomy (see Figure 3). To illustrate the expansive opportunities that these Super SDKs create for data capture, we have applied our taxonomy to two SDKs owned by Google that ASOS is using: Firebase and Admob (their global Ad Network). In so doing, we demonstrate how prominent these infrastructures are by using the taxonomy to illustrate the multiple points of contact these SDKs in particular have with the app.

Figure 2. ASOS APK inspection.
Figure 3. Expanded view of SDK services in ASOS. SDK: Software Development Kit.
Firebase SDK

Firebase exists in roughly 3 million apps (Firebase, n.d.), likely because of the broad functionality it provides across an app’s development lifecycle: building, testing, releasing, monitoring and engaging. As we can see from Figure 4, it offers a dizzying array of services that become visible using our taxonomy. Briefly, these include:

1. **Programmatic AdTech**: Firebase provides a range of analytic software to understand user behaviour via the tracking of all in-app events. These include audience segmentation tools, infrastructure to build profiles by facilitating A/B testing on anything from fonts, to messages, to ads, to links, which in turn can be applied to consolidate how users are engaged both inside and outside the application. Attribution services provide tools for tracking users, including in-app purchases, campaigns/messages that users may have seen, or retargeting efforts for partially completed purchases. While it does not provide advertising, it can connect with Admob by feeding forward its user profiles, thereby creating incentive for developers to use Firebase. From its privacy agreements, we know its services collect a range of identifiers including age, gender, location, language, interests, email addresses and phone numbers to build, consolidate and test user profiles for monetisation.

2. **Development Tools**: Firebase offers machine learning infrastructure for image and/or facial recognition or for new kinds of powerful analytics via its BigQuery tool – Google’s serverless data warehouse which can analyse at scale, petabytes of data to find insights on user behaviour. The taxonomy yields critical insights about how user data is quietly cross referenced through several different portals. Figure 5 is an analysis of Firebase’s privacy policy, for 12 of their development tools. Overall, we observe 5 of the 12 services are collecting IP addresses, five are collecting Firebase’s installation IDs, which can be used to identify a user and the Firebase Machine Learning services are potentially collecting any images or data uploaded by developers who are using their software to train with ML tools. What is unclear are the ways in which data might be shared between these services and/or with the other service clusters within the taxonomy (which we assume to be the case).

3. **App Extensions**: The most common extension that Firebase offers are the multiplatform social logins (Google, Facebook, GitHub, Twitter, etc.), which open possibilities for the sharing and use of personal data. According to Firebase’s privacy agreement, Google’s login gathers passwords, email addresses, phone numbers, user agents (what the end-user accesses) and once again, IP addresses. Other tools available might include Google Wallet found in the Android SDK, which again, can be integrated with Firebase for more productive avenues of data capture around payment and purchase analytics.

Admob SDK

Admob (Figure 6) is a portal to Google’s vast programmatic advertising network, whilst providing a range of analytic reports based on its activities. What sets the features of this SDK apart from other platforms that offer advertising is (i) the enormous reach provided by Google; (ii) its capacity to mediate a range of different AdTech actors including the integration of over a dozen ad network SDK partners like Clever Tap or App Colony; (iii) its capacity to create ads and/or integrate them into apps; and finally, (iv) its provision of different ‘brand safety’ tools which ensures some degree of accountability regarding where their ads end up and who sees them. Noteworthy are the number of SDK partnerships or complementors (van der Vlist and Helmond, 2021), raising important questions about how to make this opaque and complex world more visible and comprehensible to both end-users and policy makers in order to reduce the number of entry points for data extraction.

Applying our taxonomy first to the app ASOS, and then to SDKs located therein (Firebase and Admob) demonstrates an expansive capacity for the capture of user data, highlighting the Saas economic model in action. For us, key governance and policy questions arise given the myriad entry points for data collection that are largely invisible and unaccounted for. We see evidence of this in three key areas: (i) Firebase’s development services that are gathering IP addresses without consent are considered as personally identifiable by the Information Commissioner’s Office (ICO) in the UK and by the GDPR in Europe, and may be considered as personally identifiable if connected with other identifiers in Canada; (ii) the potential movement of personal data outside the country wherein it was produced, such as with the Huawei SDK in ASOS and (iii) a lack of guidance when it comes to SDKs combining user data gathered from one app to enrich their own databases, to build profiles, and track users across apps and devices – like Firebase, when it shares the user profiles it builds with Admob with its partners (the third parties of the third party) to maximise advertising revenues. Data protection laws may well purport to require consent when companies harvest personally identifiable data, yet significant knowledge gaps remain on how this occurs with SDKs. Policy challenges are compounded by the practical reality that users agree not just to the terms and conditions of a single app, but by proxy, to that of every single SDK and their partners therein. We therefore see our case study as an important first step in initiating cross-disciplinary research in app, platform and datafication studies.
Figure 4. Firebase Software Development Kit services.
Conclusions and future research

While Google and Apple promise a new privacy-preserving digital world, our analysis reveals that data extraction, profile building and attribution continues to grow without adequate oversight. A data monadology positions the small of the SDK as a unique vantage point for seeing the whole of the expansionary logic of the mobile personal data economy, intensifying data fi...cation by this SaaI model. Smartphones are the most prominent devices for the circulation and processing of the finely granulated data points we generate: our actions, our choices, our movements, our behaviours, our friend networks, our identities, our lives. SDKs not only bring organisational efficiencies to ever-proliferating data flows, but they also quietly offer effortless integration of advanced AI, from image recognition to natural language processing. In short, they are the technical objects ne plus ultra for the commodification of our personal data. We developed our taxonomy to offer a more comprehensive framework for understanding the contextual specificity of not only for datafi...cation, but also for mapping the infrastructures that enable platform monopolisation. We see this as a contribution for a more critically engaged analysis to better address who is capturing which data for what purpose.

The SDK taxonomy is a framework in progress. There is a need to move beyond just mobile applications, especially given that SDKs also provides software and hardware for IoT connected devices. Indeed, Google’s recently launched Cross Device SDK heralds this trend, and brings future research challenges into sharp relief (Roth, 2022). Soon users will seamlessly start a task on one device and continue it on another smart device, like a television, with their friends. This will enable heretofore impossible data flows, for profile building, inference generation, and action across all our smart devices. Then, there is the looming

Figure 5. Data gathered by firebase development services.
spectre of Generative AI and Customised GPTs, which seem destined to be diffused across the mobile ecosystem via SDKs.

More scholarship on SDKs will help to critically address some of the real challenges that relate to how infrastructures that enable interoperability through datafication are expediting both data capitalism and platform power. We hope our research inspires a more specific focus on the privacy and regulatory challenges posed by this SaaS economic model. The taxonomy offers an opportunity to engage more observable approaches that can contribute attestation to more adequately address the governance and privacy regulation challenges brought about by SDKs.

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

Funding
The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs
Jennifer Pybus https://orcid.org/0000-0002-1136-1237
Mark Cote https://orcid.org/0000-0001-6359-1627

Notes
1. While changes in the iOS now allows users to turn off tracking identifiers, there is no clear evidence that this stops SDKs from access anonymised end-user data. Research done by Kollnig et al. (2021) indicates that tracking still persists.
2. This term is often used to describe how a ‘hacker’ intentionally removes software restrictions that have been put in place by a device manufacturer to manipulate the code in unintended way.
3. This is even more urgent for apps that collect sensitive health-related data, wherein data governance and privacy issues are exacerbated by the fact that intimate data sells for 50 times more than the price of a credit data, making it extremely valuable. Between 2009 and 2019, there were 3054 large-scale data breaches that affected over 230 million accounts (Gilman, 2021).
4. See Meta for Developer’s Permission Reference list for a more complete list of what their social can access: https://developers.facebook.com/docs/permissions/reference

Figure 6. Admob Software Development Kit services.
5. There are likely more Super SDKs, especially if we look outside the Western apps we examined.

6. A more comprehensive overview of our method to open up and inspect APK files, including step-by-step workshop notes is available on https://github.com/jenpybus/SDK-Analysis

7. Huawei has emerged as an SDK in the past few years. After a cursory overview of these SDKs assets, we think it shows potential to be categorised as a Super SDK, however more research needs to be done to conclusively make this argument (Huawei, 2022).

8. We use this for its easy access, while acknowledging its limitations, and noting the invaluable Mozilla safety label critique (https://foundation.mozilla.org/en/campaigns/googles-data-safety-labels/).

9. See this link for a more comprehensive list of 500 app-events that Firebase provides: https://firebase.google.com/docs/reference/cpp/group/event-names

10. See link for some of the predefined automated user data that Firebase can automatically collect: https://support.google.com/firebase/answer/9268042?visit_id=638100873779791286-3486836828&rd=1

11. As an aside, we have not included BigQuery here because it is unclear if it has its own policy or if it is simply governed by Google Cloud’s policy.


13. For more information about privacy legislation in Canada and Europe around IP address refer to the One Trust Data guidance report: https://www.dataguidance.com/sites/default/files/gdpr_v_pipeda.pdf

14. Figures 2–6 all created with RAWGraphs (Mauri et al., 2017).

References


