The Impact of Default Mobile SDK Usage on Privacy and Data Protection

Simon Koch
TU Braunschweig
simon.koch@tu-braunschweig.de

Manuel Karl
TU Braunschweig
m.karl@tu-braunschweig.de

Robin Kirchner
TU Braunschweig
robin.kirchner@tu-braunschweig.de

Malte Wessels
TU Braunschweig
malte.wessels@tu-braunschweig.de

Anne Paschke
TU Braunschweig
anne.paschke@tu-braunschweig.de

Martin Johns
TU Braunschweig
m.johns@tu-braunschweig.de

Abstract

Are mobile app developers actively enabling data collection by advertisement and analytics companies, or are they unaware of the implications of using the provided software development kits (SDKs)? Given that the current mobile app ecosystem inadvertently involves collecting user data, which often infringes upon data protection and privacy standards, the question of the underlying reason for the permissibility of data processing arises.

We contribute to this research for both Android and iOS by performing a two-step qualitative analysis. First, we conduct a structured documentation review of five advertisement and five analytics SDKs, focusing on privacy-related information. Subsequently, we implement a set of example apps utilizing the basic functionality of each SDK. This custom utilization of the SDK allows us to perform a fine-grained traffic analysis of each required step from initialization until utilization.

Our results show that only little guidance on data protection compliance is provided. The observed network traffic shows that overall data collection by SDKs is similar between operating systems and only requires basic usage by the developer to trigger. We discover that with current SDKs, developers have minimal influence over the collected data, as merely using the basic functionality already results in data collection, with advertisement SDKs collecting more data than analytics SDKs. Overall, we explain the observed data protection infringement in ongoing mobile privacy research by documenting how developers must bear with opaque SDKs that lead to data collection simply due to usage.

1 Introduction

Since the unveiling of the first smartphone, smartphones have become increasingly popular and indispensable in everyday life. They are essential to private and business life, providing permanent and immediate access to information and entertainment or as a second authentication factor. Nowadays, there is a suitable application for all areas of life, whether tackling health problems, managing bank accounts, or interacting with others via messenger and dating services. This implies that smartphones carry a wealth of sensitive information that users prefer to keep private, as well as behavioral and location data.

> *"While many of these SDKs offer configuration options to respect COPPA1 by disabling tracking and behavioral advertising, our data suggest that a majority of apps either do not make use of these options or incorrectly propagate them across mediation SDKs."* – Reyes et al. [69]
>
> *"However, we found that the vast majority of developers did not change trackers default options that might lead to more data sharing than necessary."* – Kollnig et al. [52]
>
> *"In addition, some respondents claimed to be aware of GDPR relevant data, but were surprised by our reports which showed that the SDKs collected information."* – Nguyen et al. [58]

Figure 1: Previous findings on the privacy impact of SDKs.

Advertisement and analytics companies seek to collect as much of this user data as possible [68] threatening the privacy of the app user. These companies entice developers into their data collection efforts by offering direct monetary compensation for data points harvested through their apps [56]. Alternatively, they promise enhanced user engagement and retention rates when developers integrate their analytics libraries. The influence of these entities on the smartphone app ecosystem is ubiquitous. A large portion of available apps includes third-party tracking software development kits (SDKs), often without the knowledge of the average user [76]. Studies have shown that, as a result, apps not only share personal data with third parties [49, 52, 58] but also do so without obtaining user consent [50, 59].

The role and responsibility of app developers in these data protection and privacy violations are actively researched. Findings suggest privacy infringements result from unintentional actions by the developers [54, 58, 71] or vendors steering developers towards privacy-compromising configurations and SDKs [35, 56, 73, 75]. Still, the question of how documentation and usage patterns lead to personal information collection is open, and research needs to address it. Recent literature has highlighted the apparent gaps in developers' usage and understanding of advertisement and analytics SDKs, especially concerning privacy implications. Figure 1 showcases key statements from previous work that underscore this concern. These observations frame our motivation and lead to our investigation.

We conduct our work in a two-step process. Initially, we identify the top five advertisement and analytics providers and perform a

Proceedings on Privacy Enhancing Technologies 2025(1), 808–823

https://doi.org/10.56553/popets-2025-0042

structured documentation review to comprehend the informational material available to a developer attempting to implement the core functionality provided by the corresponding SDK. Our structured documentation review focuses on eight key questions about their SDKs' privacy impact and features. We want to understand how documentation provides information on whether and which personal data the corresponding SDK transmits, how the SDK handles consent, and if the contractual partner or their data center is within the European Economic Area (EEA) and thus stores data compliant with the data protection legislation. The second step involves implementing example apps that employ the SDKs to execute their core functionality, such as analyzing data or displaying advertisements. Subsequently, we run each app and perform an in-depth traffic analysis to analyze the transmitted data.

Our results cast light on the current state of privacy-specific information in documentation and default data collection by SDKs. Most manuals hint that usage requires consent by the user, at least by referring to "applicable law", and indicate that the SDK collects personal data. The overall specificity of how to use this information is sparse. Only six of our SDKs provide APIs to set consent, including three that provide a built-in consent dialog functionality. Overall, the manuals paint a picture of vendors attempting to protect their legal standing by referencing data protection legislation but trying to stay unspecific. Thus, vendors leave developers without a legal background or support alone to fulfill the legal requirements.

Our results from the implementation and traffic analysis support this observation. The collected traffic indicates that SDKs start collecting data early on and without explicit triggers from the developer. Thus, the developer must understand that consent and other privacy precautions must happen before initiating or initializing the SDK. We also see different implementation variations across SDKs and operating systems.
Our contributions covering both Android and iOS are:

- A privacy-implementation-focused structured review of the SDK documentation of the top five advertisement and analytics SDKs
- A minimal utilization implementation for each SDK with a separation of the different phases of SDK usage
- Traffic measurement and analysis for each app resulting in an in-depth comparison of data transmission between SDKs, operating systems, and implementation steps

We structured the remainder of the paper as follows: In Section 2, we provide information on advertisement and analytics SDKs and discuss the data protection legislation in the European Economic Area as well as the incentives for developers to include advertisement and analytics SDKs. Section 3 presents our structured documentation review, detailing our review questions, the SDK selection, and results. Subsequently, we present our technical analysis in Section 4, detailing our implementation process, followed by an analysis of the collected traffic. In Section 5, we tie our results together and discuss how our results paint a picture of opaque SDKs and naïve developers. Finally, we discuss related work in Section 7 and summarize our overall contributions and key findings in Section 8.

2 Advertisement, Tracking and its Regulations

App developers are incentivized to include third-party SDKs that offer advertisement and analytics capabilities in their applications to earn money [56]. The numbers published by Alphabet show how revenue from the advertisement has risen from 146 billion US dollars in 2020 to 224 billion US dollars in 2022 [6]. These revenues indicate the potential of personalized advertisement as conducted by Google. At the same time, they show the need for the existing data protection legislation. To protect the data subject, there are various legislation regulating the processing of personalized data, such as the General Data Protection Regulation (GDPR) [5] and ePrivacy Directive (ePD) in the EU [3, 4], the California Consumer Privacy Act (CCPA) [1] in the USA or the Consumer Privacy Protection Act (CPPA) [2] in Canada.

In this section, we discuss the incentives to use third-party advertisement and analytics SDKs (Section 2.1) and emphasize the legal framework on personal data collection applicable in the EEA (Section 2.2).

2.1 Mobile Advertisement and Analytics

In-app advertisements offer a straightforward way to generate revenue with an app, even if the app is free to use. Current statistics show increased spending on in-app advertising from 174 billion US dollars in 2020 to 272 billion US dollars in 2022 and forecast growth to 498 billion US dollars in 2028 [7]. To generate revenue from advertisements, app developers allocate portions of the app's display real estate to advertisement providers, e.g., as a banner at the bottom of the screen. Advertisement providers offer SDKs the necessary functionality to display advertisements within the app's layout. Usually, third parties provide ads to the SDK provider and pay for the ads served. The provider forwards a part of the payment to the app's publisher. It is commonly asserted that targeted ads are more lucrative than generic ones for the advertiser and even sometimes claimed to be more user-friendly for the consumer [40]. However, providers must collect a large set of user personal data to provide such a personalization. This leads to advertisement SDKs that not only facilitate displaying ads but also collect data from the devices displaying the ad.

In contrast to advertisements, *analytics* provides no direct way to make money but a means to analyze an app's usage behavior and user base. This may include general data such as geographic location and age range but can encompass more specific data such as gender. Additionally, analytics SDKs implement user behavior-tracking through event logging. Events usually correspond to typical user habits and actions, e.g., app installation, finished levels in games, or successfully sent messages. They allow developers to understand behavioral patterns regarding their apps and react to them by optimizing the corresponding sales funnels. Analytics providers offer such collection capabilities in their SDKs, aggregate the collected data, and present it to customers in dashboards. These mechanisms necessitate (personal) data collection and processing.

Past research [42, 71] has shown that developers know their users' data protection rights. According to Shilton and Greene [71], regulatory practices and development ethos differ between iOS and Android, with developers for both platforms generally considering privacy and conducting discussions about user privacy. However,

developers tend to choose the easiest solutions, which are usually less privacy-preserving, due to the need to monetize apps and the constraints of time and money [56]. Furthermore, interviews with developers about invasive data collection practices indicated that developers expect SDK vendors to keep them out of conflict with data protection regulations [58]. The question then arises: how easy or hard is it for developers to use advertisement and analytics SDKs in a privacy-preserving or at least respecting fashion?

2.2 Data Protection and Privacy Regulations

The data protection legislation in Europe consists of various laws. These laws apply not only to data processing entities that have their establishment in the European Union but also to all companies that process personal data of data subjects in the European Union, for example, to offer them services. Therefore, European data protection laws must also be observed by companies outside the EU if the data processing affects natural persons from the European Union. Past research observed that apps transmit various metadata from the app user, such as the name of the device, the user's location, the data subject's access provider, and other information [49, 50, 52, 58, 59]. Individually, many of these data points are not personal data because an individual cannot be identified through these. However, when aggregated, they can enable the identification of the user, which typically necessitates the application of data protection laws.

2.2.1 The General Data Protection Regulation (GDPR). The GDPR has been harmonizing data protection law within the European Union since May 25, 2018 [5]. This regulation is a significant milestone in data protection regulation. The substantive scope of the regulation is defined as the processing of personal data. Personal data refers to information about an identified or identifiable natural person. Since there is no insignificant data [43], the scope of this data protection law is very broad.

2.2.2 The ePrivacy Directive (ePD). While the GDPR regulates the basic principles of data processing, the EU's ePrivacy Directive continues to be applied, regulating data protection within the scope of electronic communication. The ePrivacy Directive also applies to the storage of information or access to information already stored on the end device, such as on a user's smartphone, according to Article 5(3) of the ePD. Therefore, the transfer of non-personal data may already be covered by the ePrivacy Directive. Access to all this data is only permitted with the consent of the affected user. For personal data initially collected and processed on a (telecommunications) terminal equipment and transferred to a company's backend with the user's consent, the GDPR applies to further processing outside the connected equipment.

The directive, therefore, includes sector-specific data protection requirements that take precedence over the provisions of the GDPR in this specific context [44]. However, there is no conflict where the directive does not contain specific provisions. In such cases, the aforementioned general data protection regulations of the GDPR apply. The distinction between these two legislations must be carefully examined for each area of application [4]. If data transmission is primarily related to electronic communication, the respective national data protection laws implementing the directive must be considered mainly. Data processing in this context can also be justified by obtaining consent.

While EU regulations are directly applicable in the member states, directives must be implemented through national laws in the member states. This creates a level playing field in implementing data protection law within the EU. Data protection law, a relatively young field of law, is still heavily influenced by the interpretation of the national courts and supervisory authorities.

2.2.3 Processor, Controller, and Joint Controller. Every form of data processing is affected by data protection law. This already includes the mere collection and structuring of data. The GDPR distinguishes between the data controller and the data processor. The latter can provide (technical) support to the controller in data processing. For this purpose, a contract between the parties defining their respective tasks is required (Art. 28, GDPR). The processor is subject to the instructions of the data controller. Nevertheless, both parties must independently adhere to the data protection regulation requirements. In addition, two parties can become joint controllers of data processing.

A joint controllership exists when two or more controllers jointly determine the purposes and means of processing personal data (Art. 26, GDPR). According to Article 26, they must establish an agreement in a transparent form. This agreement must outline the relationship between the joint controllers and disclose who fulfills the respective data protection duties, such as exercising the users' rights. This agreement must also be made available to the data subjects, as Article 26(2) GDPR stipulates. A unilateral transfer of all responsibilities to one party is not possible. Instead, the role of each party must be reflected accordingly. Likewise, the user remains free to assert their rights against both parties. The CJEU confirmed in its ruling that companies using technical infrastructure provided commercially by another company, which also processes personal data, are jointly controlling this data processing [61]. The ruling mentioned earlier, concerned the operation of a Facebook fan page. The company operating the fan page was deemed a joint controller along with Facebook and thus held jointly responsible for Facebook's data processing. Consequently, companies utilizing this technical capability can also be held accountable for unlawful data processing.

2.2.4 Principles of Data Processing and Consent. The GDPR establishes data processing principles (Article 5), including purpose limitation, storage limitation, and data minimization. Developers must consider these principles from the outset due to the data protection principle by design and default (Article 25 GDPR). The data controller must comply with these data processing principles and demonstrate compliance ("accountability"). Furthermore, Article 6 and Article 9 stipulate the conditions for permissible data processing for particularly sensitive data. Special categories of personal data that require heightened protection include, among others, health data or data concerning a natural person's sex life or sexual orientation. The application of Article 9 GDPR may be warranted as soon the app's name is transmitted to a third party. App names such as the queer social network "Grindr" or apps that measure blood sugar levels for people with diabetes can already imply susceptible user data.

Key processing authorizations include the users' consent and contractual agreement, provided that data processing is necessary for the execution of the contract, as well as technical or legal necessities. For consent to be valid, various formal and substantive requirements must be met. The person concerned must have the capacity to consent. A verifiable form should be chosen for the declaration. This must be given unambiguously. Furthermore, consent must be declared before the data processing and must be freely and informed given [41, Dirk Heckmann and Anne Paschke, § 7 para. 33]. Consent obtained in practice likely does not meet the extensive legal requirements [36, 50, 55, 59].

Every actor who processes personal data must meet these requirements to ensure that the data processing is lawful. A tripartite or multi-party relationship arises when SDKs are integrated into an app. The user may enter into a contract with the app operator/developer or provide consent for certain personal data to be processed by them. Additionally, the SDK provider processes data. For consent to be considered voluntary, the user must be aware of this processing and actively consent to the data processing by the SDK provider or enter into a contract with them.

2.2.5 Transfer of Data to outside the EU. The transfer of personal data to third countries outside the EU is strictly regulated and only permissible under the conditions set out in Article 44 of the GDPR. Data transfers to third countries are permitted based on an adequacy decision. With this decision, the EU Commission determines that a third country provides a level of data protection equivalent to that of the EU (Art. 45, GDPR). Various adequacy decisions for data transfers to the USA have been declared invalid by the European Court of Justice (ECJ) in the past [62, 63]. There is an adequacy decision with the "EU-U.S. Data Privacy Framework", which allows data transfers to the USA to certified companies [8]. However, a review by the ECJ is expected to follow soon.

In addition, a data transfer can be permissible under Article 46 GDPR if the recipient provides appropriate safeguards for protecting the personal data. The basis for the data transfer can thus also be the so-called Binding Corporate Rules or the Standard Contractual Clauses provided by the EU Commission. However, using Standard Contractual Clauses for data transfer to the USA was found insufficient by the ECJ in 2020 [63]. Another option for data transfer, according to Article 49 GDPR, is the existence of the data subject's consent as an exceptional circumstance. For this, however, the user must be transparently informed in advance about the data protection risks of the data transfer.

Furthermore, data transfers to third countries are permitted for hosting purposes—provided that the data is encrypted so that technically, no one in that third country can access the personal data.

2.2.6 Lessons Learned. For this work, we want to emphasize four lessons learned: *(1) GDPR and ePD are complementary:* either legislation covers data protection with the ePD focusing on electronic communication and data already stored on an electronic device, including technical data; *(2) There are data Processors and Controllers:* processors solely act on the instructions of controllers, but both parties must adhere to data protection legislation with the option of joint controllership if multiple parties determine the data processing; *(3) Data Processing may require consent:* if data is processed that is not legally or technically required, processing requires consent, with access to data under the purview of the ePD always requiring consent; and *(4) Data transfer to outside the EU is regulated:* exporting data outside the EU is only permissible if data protection can be guaranteed.

3 Privacy Review of Documentation

So far, we have outlined the motivations for using advertising and analytics SDKs and highlighted the legal framework required by the GDPR and the ePrivacy Directive. Prior work [e.g., 52, 69] has shown that developers seem to improperly configure the used SDKs when it comes to data protection. As a result, the question arises of whether the improper configurations originate from documentation that is difficult to understand, a lack of overall features not described in detail, or a reason outside the influence of the SDK providers. We conducted a structured review of the corresponding documentation to explore this question comprehensively. We first list our review questions (3.1) and then proceed to detail our methodology (3.2) as well as subsequent results (3.3).

3.1 Review Questions

For the documentation review, we consider eight *review questions (RQs)*. We deduced the first four questions directly from requirements of the data protection legislation:

- **RQ1: Legislature/Data/Declaration**—Does the SDK recommend or indicate that developers need to reference it in a privacy policy? The data protection legislation requires this as soon as the SDK processes the personal data of the user (Art. 13, GDPR).
- **RQ2: GDPR/ePD/Data/Access**—Does the documentation detail personal or any data transmission or access (Art. 13, GDPR/ Art 5(3), ePD)?
- **RQ3: GDPR/ePD/Consent/Required**—Does the SDK documentation state a requirement for user consent for usage (Art. 6, GDPR/Art 5(3), ePD)?
- **RQ4: GDPR/Data/EEA**—Is the vendor located in the EEA? Data transmission is illegal if the vendor resides in a non-EU country without complying with Articles 44 et seq. GDPR provision.

In addition to the RQs specified by both the GDPR and ePD, we identified three questions about features that make it easier to be compliant with the data protection legislation:

- **RQ5: Consent/Management**—Does the documentation describe a way to set the user consent to data collection, or has consent to be managed outside the SDK?
- **RQ6: Consent/Revocation**—Does the documentation describe a way to revoke consent given by the user? Data protection law requires that consent be as easy to revoke as it is to give.
- **RQ7: Consent/Dialog**—Does the documentation describe a utility showing a privacy consent dialog, or does the developer have to take care of this independently?

Finally, based on our motivation, in a final review question, we investigate whether SDKs simplify data collection and thus tempt a user to gather more data than necessary:

Table 1: The average rank for our selected SDKs based on rankings provided by AppBrain [18], Appfigures [19, 20], Exodus [23], G2 [26], and Statista [30, 31].

Advertisement	∅ Rank	Analytics	∅ Rank
AdMob	1.4	Firebase Analytics	3.0
Facebook Ads	1.8	Meta App Events	6.0
Unity 3d Ads	3.0	AppsFlyer	6.6
AppLovin	5.2	Amplitude	7.2
Vungle	8.6	Flurry	7.4

RQ8: Data/Extendable—Does the documentation describe or advertise how to expand the default data collection behavior?

3.2 Methodology

To find out how well the documentation guides a new user who does not know the SDK, we focused on the documentation sections that provide the required knowledge to integrate the core functionality of the SDK into an app. Our overall focus is to investigate whether the documentation addresses data protection based on the presented review questions.

Documentation Source Selection. To identify the documentation of the most popular SDKs for advertisement and analytics, we used Google Search on the 7th of July, 2023, to identify rankings for both types of SDKs. We calculated the average rank based on the identified rankings and the Exodus project [23].

For this purpose, we considered the top ten listed SDKs for advertisement and analytics from each ranking. Subsequently, we averaged the position of each SDK across all selected rankings. If one of the rankings does not rank an SDK ranked by the other, we assign it a rank of eleven. In case a provider split the ranking between Android and iOS, we used the highest reported rank of the SDK for either operating system. Finally, we selected the first five SDKs available for research, e.g., the documentation and SDK are accessible. Concerning Advertisement, we skipped IAB Open Measurement and AdColony as the first is only a meta SDK, and the second reached its end of life. The Appendix lists the individual rankings, and Table 1 lists the aggregated rankings.

Documentation Analysis. Our focus is to analyze the main documentation provided by the SDK vendor; thus, we only include the official documentation/manual of the SDK. We spent only a limited amount of resources for each piece of documentation to simulate time and money pressure and only followed the documentation until we could implement the primary use case of the SDK. Our rationale for those review conditions is based on previous research in documentation usage by developers and structured documentation analysis [51].

Due to our review set-up, we only assess the *getting started guide* and the *platform specific (Android/iOS) implementation guide* until we can implement the targeted basic functionality. The basic functionality of advertisement SDKs is to show banner ads and log events for analytics SDKs. We supplement this with a check of the navigation and referenced links for keywords indicating privacy-related content, e.g., "Privacy Policy", "GDPR", "Best Practice", and "Privacy" and exercise the search utility.

Our process uses two researchers to review each documentation, one for Android and one for iOS. In case of a mismatch between the results, the researchers discussed their findings to reach a consensus, e.g., in case one reviewer missed a detail. If the mismatch is due to the different targeted operating systems, we report our results as operating system dependent.

We designed each question with a clear answer domain to ensure comparability across the different SDKs. RQ1, RQ2, RQ7, RQ5, RQ6, RQ3 and RQ8 are binary *yes/no* questions. RQ4 adds the options that the answer is yes, but only because it is *subsidiary of a larger non-EEA located firm* or *the used servers are located in the EEA*.

3.3 Results

In the following, we report our answers to our review questions. In particular, we look at the differences between iOS and Android and across SDKs.

Reference to the Privacy Policy (RQ1). First, we investigated which SDK manuals state their vendor needs to be included in the app's privacy policy using the SDK. In total, five out of ten manuals refer to this for both operating systems. Google AdMob even emphasizes that developers have to name Google as a party that *collects and accesses* the data and provides a checklist to fulfill the requirements.

Interestingly, two SDKs mention a privacy policy requirement for only one of the two operating systems. Namely, Meta Audience Network and Meta App Events only mention the privacy policy requirements in their Apple *get started guide*, where they explicitly state that you must disclose that data is sent to Facebook in the privacy policy. We did not find a corresponding guideline in the Android version.

AppsFlyer and Amplitude unspecifically point out that the user must fulfill the necessary legal requirements.

Documentation of Personal Data Transmission (RQ2). We identified nine SDKs that report about data transmission in the documentation. This question's noticeable aspect is the varying detail level in which manuals describe their data transmission.

Requirement for User Consent (RQ3). Another important goal of our reviews was to investigate whether the documentation explicitly refers to usage requiring user consent. This is the case for six SDKs but to a differing degree of specificity, and a seventh only doing so in the iOS documentation. AppsFlyer states that a custom opt-out functionality or a restriction on the partners with whom they share their data might be required for legal reasons. Other SDKs are more specific and directly mention consent requirements in the context of privacy laws. But a common conspicuous theme is advising the developer to initialize the SDK as early as possible, "ideally at app launch", which may result in preloaded ads and transmitted data before consent. This advice commonly conflicts with any explanations on consent, stating that consent is required for using the SDK.

Vendor Location (RQ4). Vendor or processing location was not a commonly discussed data point in the manuals. We had to search

Table 2: Answers to our research questions obtained by the documentation review. ● yes, ○ no, ◑ partially (*i.e. subsidiary or server located inside the EEA*).

SDK	RQ1 GDPR/ Data/ Declaration	RQ2 GDPR/ Data/ Personal	RQ3 GDPR/ Consent/ Required	RQ4 GDPR/ Data/ EEA	RQ5 Consent/ Management	RQ6 Consent/ Revokation	RQ7 Consent/ Dialog	RQ8 Data/ Extendable
Google AdMob [13, 14]	●	●	●	◑	●	●	●	○
Meta Audience Network [27]	○ ●	○	○ ●	◑	○	○	○	○
Unity 3d Ads [32]	●	●	●	◑	●	●	●	○
AppLovin [21]	●	●	●	○	●	●	○	●
Vungle [33]	●	●	●	○	●	●	●	●
Firebase Analytics [24]	○	●	○	◑	●	●	○	●
Meta App Events [28]	○ ●	●	●	◑	○	○	○	●
AppsFlyer [22]	○	●	●	◑	●	●	○	●
Amplitude [15]	○	●	○	◑	○	○	○	○
Flurry [25]	●	●	○	◑	○	○	○	○

the linked privacy policy to determine the answer to our review question. Only two vendors completely reside outside the EEA: AppLovin and Vungle. Most others have an EEA subsidiary, and the Amplitude SDK can be configured to send data to their EU servers [34].

User Consent Setting (RQ5). Six out of ten SDKs offer the functionality to set consent in the SDK. AdMob provides the functionality to set the user consent but not to revoke it (RQ6) as it manages the user consent via their consent dialog facilities. However, it is possible to redisplay the consent dialog for the user and thus indirectly change the consent state. In the case of Meta App Events, the documentation provides information on postponing certain SDK features "to obtain user consent or fulfill legal obligations" [29] but no facility to handle consent. For the remaining SDKs, consent management is left to the developer.

Revocation of Consent (RQ6). All six of the SDKs with consent management facilities offer the option to revoke consent. AdMob provides two options: one utility is explicitly intended for development only and not meant for production code, and the second option is to redisplay the consent dialog for the user. Direct access to the consent value is not possible.

Privacy Consent Dialog (RQ7). Only three out of ten SDKs directly ship with an internal mechanism for displaying a consent dialog. AdMob, in particular, explains in detail for both operating systems how to display such a dialog for both operating systems. Unity3D states that it displays a consent dialog before showing the first ad, similar to Vungle.

Data Collection Extension (RQ8). AppLovin and Vungle are the only SDKs for advertisement that allow for the extension of data collection. Developers can provide additional data fields to supply more information about their users, such as gender, phone number, email address, and interests. In contrast, AdMob has a list of deprecated features indicating that they used to allow data collection, such as gender and age, but they have phased out this practice. In the case of the analytics SDKs, three offer the functionality to extend data collection. For example, AppsFlyer lists a "Customer User ID" to cross-reference users across sessions, whereas Firebase Analytics can log additional events containing arbitrary data. All Analytics SDKs can log custom events with custom values, but those would not carry any meaning for the SDK provider.

4 Instrumentation & Traffic Analysis

We have explored the documentation of the different SDKs and now want to analyze and compare the data transmissions of the SDKs. To achieve this, we will conduct an in-depth network traffic collection study for the selected SDKs. We created a sample app for each SDK on both iOS and Android. We then executed each sample app and monitored and analyzed the network traffic.

We start with a detailed implementation description for each of the analyzed SDKs (4.1). Continue with our measurement setup (4.2) and end with a presentation of the observed SDK traffic (4.3).

4.1 Implementation

We created sample applications for each SDK in our implementation dataset on both iOS and Android. To ensure the correctness of our implementations, we followed the documentation given by the SDK provider, as examined in Section 3.

We aim not only to understand the data transmitted while using the SDKs but also to understand at what point the SDK transmits what kind of data. To this end, we split our proof of concept apps into four steps, presenting the intersections of minimally required functionalities across the selected SDKs:

(1) **Creation phase.** Some SDKs require the creation of an object, be it by calling the factory function to retrieve a singleton used later on or to create an object that the app has to keep track of.

Table 3: Implementation overview for the four phases across our SDKs. The symbol ● means the functionality was present and implemented by us, while ○ means that the SDK does not provide such a functionality. *For Meta App Events, Firebase, and AppsFlyer the Init. phase happens before the Consent phase.

SDK	Creation		Consent		Init.		Util.
	🤖	🍎	🤖	🍎	🤖	🍎	
AdMob	○	●	●		●		Banner
Facebook Ads	○	●	○		●		Banner
Unity 3d Ads		○	●		●		Banner
AppLovin		●	●		●		Banner
Vungle		○	●		●		Banner
Amplitude		●		○	○	●	Event
Flurry		●		○		●	Event

SDK	Creation		Init.*		Consent*		Util.
	🤖	🍎	🤖	🍎	🤖	🍎	
Meta AE	○	●	●		●		Event
AppsFlyer		●	●		●		Event
Firebase	●	○	○	●	●		Event

(2) **Consent phase.** Some SDKs supported either storing the user's consent to data collection or allow disabling data collecting and reactivating the collection after the app has acquired consent.
(3) **Initialization phase.** Most SDKs require some form of initialization by calling an initialization function.
(4) **Utilization phase.** For advertisement SDKs, we chose to display a banner ad as our basic functionality. For Analytics SDKs, we explicitly log an event.

As each SDK varies slightly in its setup and implementation, we give a summary of the implementation details for each phase Table 3 gives an overview of the implementation steps across our sample apps.

Presence of Phases. While AppLovin and AppsFlyer provide the expected four phases for both operating systems, AdMob, Firebase, and Meta App Events have each phase for at least one operating system. Four SDKs lack a phase for Android, iOS, or both. Notably, the consent phase is missing in three SDKs entirely, followed by the creation phase, which is not present in Unity3D Ads or Vungle. In our implementations, we skip a given phase if it is not provided by the SDK.

Order of Phases. All SDKs but Meta App Events, AppsFlyer, and Firebase expected consent before initialization. We adapted the order of phases accordingly.

Creation. If an SDK provided some form of object creation mechanism, usually via *getInstance*-type getter for an SDK object, it was predominantly either only for iOS (three SDKs) or both operating systems (4). Among the three remaining SDKs, one provides such a mechanism only on Android, while the last two remaining SDKs (Unity3D Ads and Vungle) provide no such mechanism on either operating system.

Consent. Seven out of ten SKDs allowed for some form of consent configuration. AdMob provided an on-demand consent dialog functionality, while Unity3D Ads and Vungle only provided an indirect consent dialog functionality (comp. Section 3). Their documentation states that the consent dialog automatically shows before the first advertisement. Either SDK also allowed for manual consent management, and we opted to use that option to ensure comparability with the other SDKs. AppLovin only provides a manual configuration function to set consent. The remaining SDK with some form of consent option—Meta App Events—allows to deactivate data collection in the app configuration and then later on manual reactivation. A similar setting is also provided for Firebase in addition to their consent facility.

Initialization. Only one SDK—Firebase—does not provide any initialization functionality for Android. The corresponding initialization functions on iOS set configuration values that the SDK already requires during creation on Android.

Utilization. We chose to either display a banner advertisement for advertisement SDKs or explicitly log an event for analytic libraries as our utilization of the corresponding SDKs. Every SDK provided the required functionality.

4.2 Traffic Analysis

We conduct our traffic collection using the framework published by Koch et al. [50]. It uses a man-in-the-middle proxy (*mitmproxy*) [57] to intercept all requests while instrumenting each app using Frida [67]. The framework deploys Objection [70] to bypass SSL pinning on a rooted Android and leverages SSL Kill Switch 2 [39] on a rooted iOS. We use a Google Pixel 6a with Android 13 and an iPhone 8s with iOS 14.5.1 as we require *checkra1n* [10] to *jailbreak* the iPhone, which does not support newer OS versions. The framework automatically installs and runs the tested app with all required permissions granted.

We developed a plugin to guide us through the process of initiating each phase of our test apps, with each phase receiving 60 seconds of traffic measurement time. Depending on the presence of a consent option, we perform one or two independent measurements, giving and denying consent to data collection. For the evaluation, we adapt the code published by Koch et al. [48, 50] to parse the intercepted traffic and extract the different data values transmitted. As we develop an independent app per SDK, we can isolate the corresponding communication endpoints to identify the SDK-related requests. We remove requests not related to the SDK under test, i.e., conducted by the operating system. Given our in-depth qualitative approach, we inspected each encountered request manually and expanded the work by Koch et al. [50], which provides a parsing implementation for tracking and advertising endpoints, with missing parsing steps or missed data values.

We group the individual data values according to Table 4 in which we aggregate the different observed data types based on commonality. Furthermore, we attribute the different groups to the legislation that applies to them individually. All the observed data points are covered by the ePD (ref. Section 2.2) as they are

Table 4: Our grouping of collected data types into larger type groups and the legislation that applies to anybody processing its contained values.

Group	Included Values	Leg.
Stor. & Mem.	memory and storage values	ePD
Meta Data	phone meta data (e.g., maker)	ePD
Display	screen information (e.g., height)	ePD
App Data	app version, app ID	ePD, GDPR
Network	network data (e.g., IP or Carrier)	ePD
Identifier	identifiers such as GAID or IDFV	ePD, GDPR
User Agent	user agent	ePD
Hardware	headset, model, cpu, is emulator	ePD
Locale	language, time zone, country	ePD
Positioning	rotation x/y/z	ePD
Movement	accelerometer x/y/z	ePD

Figure 2: The amount of requests by each SDK across the different implementation steps.

retrieved from the electronic device, i.e., the underlying smartphone. Additionally, *App Data* and *Identifier* are also covered by the GDPR. App Data is covered by the GDPR as soon as the underlying app is sensitive, e.g., sexual orientation or health-related, as transmitting the name of the app also reveals corresponding sensitive details about the user. For example, being a user of Grindr— queer dating app—indicates a queer inclination. Identifiers are covered by the GDPR as they allow re-identification of a user and tracking them. In addition, the GDPR must be observed when it comes to regulatory content that goes beyond the ePD. This applies, for example, to the question of who is responsible for data processing.

4.3 Observed SDK Traffic

We observed a diverse set of requests across different phases in our SDK instrumentation and a broad set of data points accompanied by identifiers. Figure 2 visualizes the observed amount of requests per phase and Figure 3 the in-depth traffic analysis and extraction of data points.

4.3.1 Collected Requests. Overall, we collected 199 requests with more requests on iOS (123) than on Android (76). We were unable to break the SSL connection for 2 requests on Android, without any such problems on iOS. For both Android and iOS AppLovin was the SDK with the most requests, while Amplitude for Android and Meta Audience Network for iOS are the SDKS with the least amount of requests. Table 5 gives an overview of the intercepted requests aggregated across SDKs.

When analyzing requests based on the phase, we observe that SDKs on Android had on average 19.0 requests and iOS SDKs 30.75. The standard deviation is high with 14.21 for Android and 23.55 for iOS. Both systems agree that SDKs perform the least amount of requests during creation and the most during utilization. Figure 2 visualizes the request distribution across SDKs and phases. The figure displays the number of requests split across phases and SDKs, with larger dots signifying more observed requests.

We observed mismatches in contacted domains for all SDKs but Google AdMob when comparing Android with iOS and consent to no consent. In Firebase Analytics, Flurry, and Amplitude we observed that either iOS or Android contacted an URL earlier than the other operating system, but when looking at the whole set of contacted URLs the SDKs agreed across operating systems. The remaining SDKs exhibited more pronounced mismatches. Meta App Events, Meta Audience Network, AppsFlyer, and Unity3D on Android and iOS visited different subdomains while the set of domains itself stayed the same. Meta App Events visited one new domain on iOS fbcdn.net whereas Android only visited graph.facebook.com, which the SDK contacted on both Android and iOS. AppsFlyer and Unity3d both visited different subdomains of their domains appsflyersdk.com and unity3dusercontent.com, unity3d.com. Finally, AppLovin and Vungle visited not only different subdomains but also external domains that were not directly related or owned by the vendor to our knowledge. We suspect that this is due to the auctioning process of ads, and them being loaded from different locations depending on the bid. The only SDK that was consistent in contacted domains is AdMob.

4.3.2 Observed Data Transmission. We were able to break the SSL connection and analyze the content of observed requests for all but two requests on Android, all belonging to our App for Google SDK AdMob. When analyzing the decrypted traffic, we observed deliberate obfuscation by AppsFlyer and AppLovin, which we were only partially able to break for AppLovin. Consequently, our presentation of observed data transmissions focuses on the remaining seven SDKs. Figure 2 provides an overview of the transmitted data during the different phases while Table 6 quantifies the data points transmitted, encompassing a summary comparison across all phases. The figure excludes AppsFlyer due to the unbroken obfuscation but still includes AppLovin.

The phase with the most data transmission by the SDKs was initialization, with an average of 13.57 data values on Android and 15.86 on iOS. Creation was the phase with the least amount of transmitted data values, as only Flurry transmitted data values and only on Android. Each phase exhibits differences in the types of data values transmitted by the SDKs when comparing Android

Table 5: Overview over the observed aggregated SDK related requests and our ability to successfully break SSL Pinning when giving consent.

	Total	Failed	Min	Max	∅
	76	2	1	33	7.6
	123	0	2	44	12.3

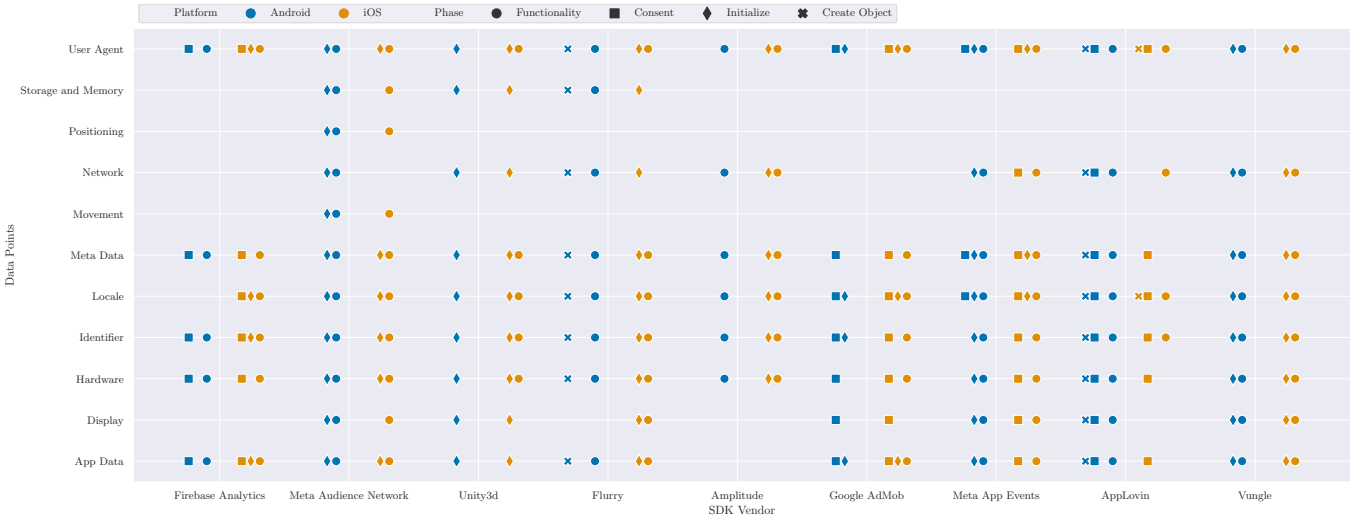

Figure 3: Visualization of the different data points collected by each SDK during the different instrumentation phases with consent.

Table 6: Unique data values transmitted during the different phases in all SDKs for which we had a complete understanding of their traffic. Δ describes the amount of data types that were different across Android and iOS.

SDK	Creation			Consent			Init.			Util.			Σ		
			Δ			Δ			Δ			Δ			Δ
Meta Audience Network	0	0	0	0	0	0	26	8	5	24	21	1	34	23	1
Unity 3d	0	0	0	0	0	0	40	46	0	0	9	0	40	46	0
Vungle	0	0	0	0	0	0	18	21	0	18	27	0	20	27	0
Firebase Analytics	0	0	0	8	9	1	0	4	4	7	9	1	9	10	1
Meta App Events	0	0	0	3	12	5	11	4	5	11	11	0	11	13	0
Amplitude	0	0	0	0	0	0	0	10	6	8	10	0	8	10	0
Flurry	20	0	8	0	0	0	0	18	9	20	13	3	23	18	1
⌀	2.86	0	1.14	1.57	3	0.86	13.57	15.86	4.14	12.57	14.29	0.71	21	18.71	0.43

with iOS. Initialization exhibited the largest discrepancy with an average of 4.14 data types across the SDKs. When examining the amount of data values and types transmitted across all phases, both operating systems barely differ. They only exhibit an average discrepancy of 0.43 values across the SDKs. The difference in data types is due to the fact that Meta Audience Network collects Device Network information on Android but not on iOS, Flurry collects Device Display information on iOS but not on Android, and Firebase Analytics collects Hardware and Metadata information on iOS. All other SDKs collected the same data types across both operating systems.

Notably, all SDKs collected some identifiers with their first and subsequent requests. All but Amplitude collected the App Name. Overall we observed an identifier in requests to 94.79% endpoint URLs of our test apps, accompanying on average 8.43 ($\sigma = 11.85$) different data points. 12.50% of the identifiers were operating system identifiers such as the GAID, IDFA, or IDFV provided by Android and iOS. The remaining identifiers were UUIDs contained in the traffic or explicitly named id or identifier in the request payload.

4.3.3 Missed and Obfuscated Requests. We were unable to break SSL for 2 requests on Android, both of them transmitted by Google AdMob. The missed requests occurred during utilization. For AdMob, the SDK transmitted no additional data type on iOS when compared to Android.

We observed 38 obfuscated values for AppLovin and 2 for AppsFlyer on Android. Out of the obfuscated values, we were able to break the obfuscation for 17 AppLovin values by partially reversing the Android SDK. The deployed obfuscation is encryption using a constant byte array compiled into the Android SDK and the SDK key assigned to the app as a symmetric key. Figure 3 contains the visualization of the data points transmitted by AppLovin. AppLovin starts transmission of data values across 8 different types in the creation phase, and then constantly retransmits the same types during consent and utilization. While we were unable to break the obfuscation on iOS, we still observed 46 and 3 obfuscated values for AppLovin and AppsFlyer.

4.3.4 Transmission Without Consent. Only six SDKs provide an option to actually set consent and not only toggle data transmission,

namely AppLovin, AppsFlyer, Google AdMob, Firebase Analytics, Vungle, and Unity3D. We were thus able to observe the effect of that functionality by performing data collection and declining the consent, with the overall impact of denying consent being minuscule. AppLovin is still transmitting data values on creation, i.e., before setting consent. AdMob and Unity3D do not change their overall data transmission behavior and start transmitting data during the consent and initialization phases. Finally, Vungle starts transmitting earlier, i.e., during consent, but not during initialization anymore. Analyzing the transmitted data points and contacted endpoints, we observed more requests when we provided positive consent with Unity3D and AppLovin contacting three and five additional subdomains of unity3d.com and applovin.com with an additional visit to view.adjust.com for AppLovin. When we gave no consent, AppLovin contacted two new subdomains of applovin.com and an additional request to app.adjust.com. Vungle showed the largest differences with seven domains visited with consent but not without and six domains visited without consent but not with. We did not detect any difference in transmitted data values.

One curious artifact we observed when analyzing the no consent data transmitted by Vungle were values called *user_lat* and *user_long* sent to imp.control.kochava.com on iOS. The names indicate that those data points are related to the user. However, when we checked the coordinates (43.6171, −72.9636), they were located in the USA, whereas the phone was located in Germany during the measurements. We suspect that this is due to the refused consent but cannot verify this suspicion as we did not observe the same request when we consented to data collection (ref. 4.3.1).

The only SDK that exhibited a meaningful difference in the transmitted data and outgoing requests was Firebase. On iOS less data points were transmitted, and on Android, we did not observe any outgoing requests. We repeated the measurements for Firebase three times with a delay greater than one hour between measurements and consistently did not observe outgoing requests on Android. Its documentation indicates that Firebases does cache [12], and we cannot conclusively rule out that the lack of observed requests is due to caching utilizing the Android play services on the Pixel that are lacking on the iPhone.

5 Opaque SDKs and Naïve Developers

Our documentation review and traffic analysis provided in-depth insights into the impact of advertising and analytics SDKs on data protection. In this section, we first discuss the problems unveiled by our documentation review (5.1). Subsequently, we look at the SDK data transmission behavior (5.2) and consent configurations (5.3) and discuss the encountered obfuscations (5.4). This leaves us with a short summary of the differences between the SDKs (5.5) and three lessons learned concerning the state of documentation, data transmission, and consent (5.6).

5.1 Manuals Lack Details on Privacy

Overall, explanations and details on the privacy impact of a given SDK and the corresponding implications are sparse. Even if the documentation lists potentially collected data points, it does not specify when these are collected in the implementation process. This left us guessing on what aspects of an SDK are safe to use before acquiring consent. Especially considering that many SDKs recommend initializing the SDK as early as possible in the app start. Furthermore, while all but two SDKs state that they need to be included in the privacy policy and all but three state that consent is required, the information is not always explicit and often hidden in legalese, such as the terms of service or explicit GDPR sections. We are not convinced that a developer would even read those rather dry texts. Data transmission before consent is well documented by now and supports this theory [49, 50, 52, 58, 59].

It was difficult to figure out the geographic location of the SDK's legal entities (RQ4) or server location. We usually extracted the corresponding information from the terms of service. While this technically answers the question, we do have doubt on whether a developer will find it, and thus, we expect issues concerning data exchange outside the EU to lead to data transfer to countries without an adequacy decision on the one hand. This is an issue for the US in particular as the history [47, 64] and future [60] of the US-adequacy decision are questionable with a large portion of tech firms being headquarters in the US. One example underlining the complexity of localization is Amplitude: You can create an account at analytics.eu.amplitude.com/signup, which assigns you an EU data center, or create an account at analytics.amplitude.com/signup which does not [16, 17]. We had to investigate the Amplitude support forum discussions to understand this difference.

Finally, the frequency with which SDK manuals expect the user of the SDK to ensure compliance with the law was high. While a minority of SDKs state that privacy policies have to include them, most simply refer to "applicable law" and leave the burden of figuring out what this means to the reader. In another case, AppsFlyer stated that due to *their understanding of the Apple policy*, AppsFlyer "doesn't track", adding to the confusion.

Past work by Nguyen et al. [58] documented that developers expect the provider of the SDK to solve the question of legal compliance. This expectation conflicts with the documentation we encountered. Based on our observations and understanding of the data protection legislation, especially Art. 26 of the GDPR, we presume the developer and SDK provider to have a joint controllership on the data processing. The consequence is that either party is responsible for the data processing, with the user having recourse against either in case of data protection violations, with freedom of choice against whom to litigate [61]. This represents a risk for the developer, who cannot influence the SDKs they use for monetization or their implementation of data collection. A risk we do not see reflected in the documentation. From a privacy advocate perspective, we do consider the SDK provider to be responsible for at least providing their customers with detailed information. But we also view developers as naïve when they use SDKs that do not provide sufficiently clear information, which leaves both parties as reasons for the ongoing privacy violations in apps.

5.2 Observed Data Transmissions

Our traffic analysis shows that advertisement SDKs collect more data points than analytics SDKs. This was counter-intuitive to us, as advertisers do not need any information to serve a simple ad, whereas analytics inherently collects data. The data points harvested by SDKs are primarily about the device. While such

data values are not covered by the GDPR at first sight, they are covered by the ePD, which still requires informed user consent to process them. This predominance of device data surprised us, as these data points do not help in personalizing or improving advertisements. Our theory is, that the vendor uses the device data for fingerprinting, a possibility already demonstrated on the web by Boussaha et al. [38]. This would allow them to identify users across different apps without the need for global advertisement identifiers. Such a scenario is explicitly covered by [65] addressing device fingerprinting in the context of Art. 5(3) of the ePD. The opinion reiterates that access to any information stored on the device requires consent, even in the case of read-only values via the OS API. Thus, any such data access, especially for fingerprinting, in the absence of explicit user consent would run afoul of current data protection legislation.

All SDKs included some identifier in their traffic. This is an important observation, as data values associated with an identifier become personal data as they relate 'to an [...] identifiable natural person' (Art. 4(1) GDPR), consequently putting the data under the purview of the GDPR. All SDKs but Amplitude included the App name in their traffic. App usage by itself can imply sensitive information such as sexual preferences or health. For example, transmitting the app identifier for a dating app or diabetes management app would automatically provide information about a person's sex life or health. Arguably, this data is already leaked to the provider by the app contacting the SDK endpoints, as each SDK requires registration and usage of an assigned unique key. Thus, the provider can map requests to apps anyway. However, we observed SDKs initiating communication with third-party endpoints, which then can transmit this information to initially unaware third parties. Developers, thus, need to consider this when initiating communication with anybody who can directly or indirectly associate the communication with a specific app. Processing of sensitive information is under special protection and consideration under the GDPR (Art. 9 GDPR).

As previously indicated, we did observe differences in contacted URLs when comparing operating systems. Those diverging URLs are usually owned by the SDK vendor themselves, and we do not consider this indicative of diverging behavior between operating systems. It is sensible to provide different resources for different operating systems. The exceptions to this observation are AppLovin and Vungle initiating requests to third parties. This indicates that both before and while an ad is displayed, data is transmitted to possibly multiple different entities. An observation that aligns with concepts such as the real-time ad bidding process. However, this behavior makes data protection even harder for developers, and informed consent is nearly impossible.

Concerning the ability to collect additional data, we only found a few options in advertisement SDKs. Google AdMob even actively moved away from the corresponding API. We consider this to be a positive sign, as this implies that advertisers are not actively pushing developers towards collecting more personal data of their users. Analytics SDKs present a reversed picture, as two SDKs provide such functionality. This makes sense, given that app developers want to understand their user base, and analytics SDK providers want to fulfill this desire. However, using this API is up to the developer and is not a requirement for simply logging events. Therefore, the reason for any observed extensive data collection [49, 50, 52, 58, 59] can be found at the developers when it comes to analytics. At least if it exceeds our observed data points.

5.3 Six SDKs With Consent and No Effect

Our documentation analysis raised the question of whether an SDK requires user consent (RQ3), provides facilities to display consent dialogs (RQ7), or at least manages consent (RQ5). The results deepen our understanding of the current state of consent-related privacy infringements in mobile apps. Only three SDKs provide the utility to display a privacy consent dialog, i.e., AdMob, Vungle, and Unity3D. AppLovin's documentation states the requirement to set consent while implementing the targeted functionality. Most manuals only refer to "applicable law". While AdMob does also document how to retrieve consent via their built-in consent dialog functionality, this is not presented as strictly required by the documentation.

Meta App Events provides the ability to deactivate data collection. We consider starting and stopping data collection to be an indirect consent management. This puts the burden to ensure proper data collection on the developer, who is not familiar with the underlying SDK. Additionally, deactivating data transmission as a whole is not a meaningful choice for the developer. There is a reason a developer wants to use a given SDK. No data transmission at all would render the SDK useless, tempting the developer to skirt data protection. This state of affairs demonstrates a general lack of consideration, underlined by the fact that every SDK eventually transmits information. While we think that an SDK does not have to ship with a consent dialog functionality, a mandatory consent setting with subsequent impact is a must-have. However, our data indicates that this lack of functionality may not be the only issue regarding consent. Our observed traffic indicates that the present consent functionalities had no measurable impact on the transmitted information, except for Firebase. The vendor may treat a lack of positive consent differently on the server side, but overall, we consider this lack of a measurable effect concerning and in violation of the data protection legislation. Those results do fit observations of previous research [50, 59].

5.4 Questionable Obfuscation

Eight out of our ten measured SDKs send their traffic predominantly as plain text over HTTPS. This includes the market-dominating vendors Alphabet and Meta. The main difference is the form of the data encoding, as we encountered simple query parameters, JSON, and protobuf. Only AppsFlyer and AppLovin break with this pattern and use custom functionalities to turn large portions of the transmitted data unreadable before sending it. AppsFlyer does this for every request and AppLovin for most, with some requests still containing readable information in the query. We were able to reverse one out of possibly three different obfuscation routines for AppLovin on Android. The routine implements an encryption function that uses an app constant value as a symmetric key. Consequently, it is possible to decrypt the traffic after reversing the underlying routine. We consider this an obfuscation and not an additional encryption layer due to its ineffective protection against dedicated decryption and its symmetric key shipped with the app. As the data transmitted in the obfuscated data does not differ in

quality from the data sent by the competition, we see no strategic reason for this implementation choice. However, the negative impact of this on anybody who wants to understand the data transmitted is large. Each SDK version can potentially vary aspects of the key or encryption routine. Consequently, the obfuscation becomes an obstacle to privacy research and tooling. Each App using this SDK has to be individually decompiled to extract the static key, making large-scale and app-independent analysis infeasible.

5.5 SDK Differences in a Nutshell

The most complete documentation, in our opinion, is of Unity and AdMob, as they cover every question we posed during our review. Especially addressing Consent Management and explaining how to get consent from the user using the SDK is not a given across our reviewed documentation. This is particularly noticeable for analytics SDKSs, as only two discuss consent requirements even though they collect personal data. We found some mention of personal data collection in all SDK documentation except the Meta Audience Network. It is notable that iOS developers are provided slightly more information when it comes to Meta-provided SDKs.

The differences in the implementation are only minor, with the largest observable impact for Flurry, which already transmitted data during creation on Android but waited to do so until initialization for iOS. While this makes transitioning between OS harder for developers, the overall impact is small as the SDKs are eventually transmitting the same data points. Six SDKs provided consent management capabilities with only a minor effect on the transmitted data. The only SDK that exhibited a meaningful difference when consent was refused was Firebase. While initial data transmissions vary between operating systems and SDKs, two observations hold: I) Eventually, the SDKs predominantly transmit the same data types, and II) subsequently keep retransmitting them. This similarity of the claimed to be privacy-focused iOS [9] and Android is disappointing from a privacy advocacy perspective, but it is in line with previous research comparing Android and iOS [50, 52].

The main difference we observed is the number of requests that were higher on iOS. This difference mainly manifested in three SDKs: Vungle (16 more), AppLovin (11 more), and Unity3d (17 more). We observed those additional requests mainly during utilization for Vungle and AppLovin, and initialization for Unity3d. Those phases coincide with the time the ads are being loaded, as Unity3d seems to preload ads during initialization.

5.6 Lessons Learned

In the course of our research, we gained three lessons learned. Our *first lesson learned* is that the available documentation is not sufficient to ensure privacy-preserving usage and relegates compliance to the developer. The *second lesson learned* is that data transmission goes beyond the obvious needs of the promised functionality. This leads to our final and *third lesson learned*, that slightly more than half of the SDKs provide the ability to manage user consent with indiscernible impact. Overall, our research on the documentation and implementation leaves us conflicted. While we did gain a better understanding and sympathy for developers failing to construct privacy-respecting monetized apps, we also have to question their choice of monetization partners. None of the SDKs left us with the impression that they enable the developer to program a privacy-respecting app. As we consider it common knowledge that advertisement and analytics companies collect user data, we argue that a developer cannot feign ignorance when choosing a monetization partner. This leaves us with the question of why our selection of SDKs is chosen by developers.

6 Ethics & Limitations

Our structured document analysis runs the risk of human bias. To mitigate the risk, we used two independent researchers for each documentation, one on Android and one on iOS. Finally, we also designed our questions to ensure a clear answer domain, forcing concise answers. While our study design reflects the reality of development, we miss information provided in secluded documents or by third parties and thus present a realistic yet incomplete picture of the presented information.

As we assumed personal data collection by each of the investigated SDKs, we did not upload our apps to the app store. Collecting external personal information would put significant legal burdens on us to stay compliant with the data protection legislation. This decision can impact the observed traffic and SDK behavior, as we are unable to fully configure the advertisement dashboards that inquire about an App Store link. Facebook Ads and Vungle allowed us to enable a test mode for our apps in the dashboard to run the SDK in a realistic test mode with real content enabled [11] but without any income for displayed ads. We deploy as realistically as possible and disable any test or development modes in the advertisement SDKs. We minimized the impact on the vendors by limiting our testing and measurements to a bare minimum after deactivating test or development modes, and will not withdraw any money earned.

AdMob deploys hardened SSL connections for some of their traffic, and we were unable to break it using Objection on Android. But we were able to intercept the corresponding requests on iOS and, thus, have insights into the transmitted traffic. Extended SSL hardening is a known issue for traffic interception studies on smartphones [66]. Furthermore, AppLovin and AppsFlyer added obfuscation on top of the used SSL encryption. We were only partially able to reverse the obfuscation, limiting our understanding of the transmitted information.

Three out of the four researchers have some experience developing Android apps in the past, with the fourth being new to the app development domain, but all researchers involved in the document review and implementation are experienced programmers. While we are confident that we are sufficiently familiar with SDK documentation and adaptation for programming projects in general, our lack of commercial app development experience limits our results as we could have missed documents or implementation patterns deployed by seasoned app developers, basing our results. Future work should proceed on this avenue and involve seasoned app developers to understand how SDK documentation affects their app implementations and the subsequent data collection behavior.

7 Related Work

We split the past work into two categories: work that analyzes apps directly to understand privacy-invasive behavior and work focusing on the developer as the cause for privacy-violating behavior in apps.

Technical analysis. Considering the significant impact of the GDPR on the legal landscape of privacy within the EEA, our overview will primarily focus on works published after 2018.

Nguyen et al. [58] have used dynamic traffic analysis on Android only to show that apps are leaking personal information to third parties post-GDPR, with a consecutive developer interview revealing that developers often consider the third parties to be responsible. Work by Kollnig et al. [52] has shown that personal information leakage through apps is not different between iOS and Android devices, despite Apple's advertisement for being the more privacy-preserving choice when it comes to smartphone vendors. Koch et al. [50] made similar observations concerning personal information leakage in combination with the presence and user-unfriendly design choices of consent dialogues matching the Android-only results by Nguyen et al. [59]. Koch et al. [49] also leveraged dynamic traffic analysis to match privacy labels on the Apple App Store against the actually transmitted personal information, discovering mismatches. Further work into personal information transmission by Han et al. [45, 46] showed that paying for apps does not necessarily imply a more privacy-preserving behavior with a significant amount of apps still transmitting personal information to third parties the same as their non paid counterparts and Reyes et al. [69] discovering that the same observation holds if the apps target children despite them being stronger protected against privacy-invasive behavior by privacy laws. Son et al. [72] analyzes the software stack of popular mobile advertising libraries on Android to investigate how they protect users from malicious ads and what information the ad can gain about the user or the device. Binns et al. [37] perform a static analysis on 995000 apps on both the US and the UK Google Play Store and study the prevalence of third-party tracking across categories and jurisdictions.

Human analysis. Tahaei et al. [73, 75] looked into documentation and developer usage of said documentation. They performed a walk through four highly popular Android ad networks with the developers to integrate an interstitial ad and found diverging details concerning privacy information content and presentation across the documentation. They also looked into the effect of nudging developers towards using non-tracking ads by explaining the privacy consequence, observing an eleven-fold increase in non-personalized ads by developers [74]. Similarly, Kollnig et al. [51] analyzes common advertisement and analytics SDK documentation concerning the presence of a consent dialog and consent requirements in the provided documentation. However, they do not perform an implementation and subsequent traffic analysis of the studies SDKs to observe the transmitted data with and without consent. Mhaidli et al. [56] performed surveys and semi-structured interviews with developers and discovered that developers often leave advertisement SDKs in their default, more privacy-invasive, configuration despite claiming to be sensitive towards the privacy needs of their users, a claim also observed by Ekambaranathan et al. [42] though in constant conflict with perceived market pressure by developers. Balebako et al. [35] observed in their developer user study that especially smaller companies were least likely to engage in privacy-protecting behavior and put forward the need for tools to help developers in this regard. A call picked up by Li et al. [53] developing COCNUT, an IDE plugin warning developers against potentially privacy invasive code based on preceding interviews with 18 Android developers. Shilton and Greene [71] conducted a study of developer discourse in forums concerning privacy discussions. The results of the study showed diverging work practices between iOS and Android developers, as well as the recognition of privacy as a value in mobile development. Li et al. [54] made similar observations in an analysis of the Reddit board /r/androiddev/, with Android developers only rarely discussing privacy on their own, but actively doing so if externally triggered. Although the cited works have enhanced our understanding of app privacy behavior and the widespread adoption of advertisement and analytics SDKs, none have performed a technical, in-depth analysis of the traffic transmitted during the stages of utilizing these third-party SDKs. Additionally, past work was primarily focused on Android without a comparison with iOS.

8 Summary & Conclusion

We have presented a structured documentation review of the five most popular analytics and advertisement SDKs for Android and iOS. Our focus was on determining the extent to which the documentation assists in building applications that respect privacy and comply with data protection laws. While the documentation does provide some guidance, it often lacks specificity and frequently defers to "applicable law" instead of offering clear advice. This ambiguity left us, and possibly developers, on our own to interpret requirements, likely contributing to the current state of mobile privacy. We then implemented apps for each SDK using their core functionalities: a banner ad for advertisement and event logging for analytics SDKs. We then monitored the resulting traffic for the different applications. During our study, we observed 199 requests and noted the transmission of 51 distinct data points. The majority of this information consists of device-specific information. We observed no meaningful differences between iOS and Android data transmission patterns. Although the data our chosen SDKs transmitted is subject to data protection legislation, due to the contained identifier, we do not consider the data inherently private at first glance. However, we recognize a significant risk: vendors can abuse the diverse data points related to the device to create a fingerprint, facilitating cross-app tracking of a user. Finally, we studied the impact consent choices make on the transmitted data in the SDKs and observed no difference in the transmitted data considering any consent choice users could make.

In conclusion, our findings indicate that even the basic usage of advertisement and analytics SDKs results in data collection, potentially in violation of the current data protection legislation. It appears that users of the SDKs only have minimal influence on the transmitted data, leaving them with the unpleasant choice of either using the SDKs or avoiding them completely. Considering the essential nature of advertisement and analytics as monetization solutions for developers—as highlighted by Mhaidli et al. [56]—it is imperative to devise solutions minimizing data collection while retaining comparable features. But for now, developers have to take responsibility to avoid choosing SDKs that are not privacy-focused and do not provide sufficient information and means to develop a privacy-preserving, but economically viable app.

Availability

All code is publicly available on GitHub: https://github.com/Impact-of-Mobile-SDK-Usage-on-Privacy, including iOS and Android apps, our App Analyzer plugin, as well as the parsing and plotting code.

Acknowledgements

We used the paid services of LanguageTool[1] and Grammarly[2] for spelling and grammar checks, as well as for the occasional stylistic improvements on pre-existing text across the paper.

This work was funded by the European Union's Horizon 2020 research and innovation programme under project TESTABLE, grant agreement No 101019206, and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2092 CASA – 390781972.

We also want to thank our colleagues Jannik Hartung and Tobias Jost who helped us with reversing the obfuscation and Martin Degeling who provided valuable feedback on our research approach.

References

[1] California consumer privacy act (ccpa). URL https://oag.ca.gov/privacy/ccpa.
[2] Consumer privacy protection act (cppa). URL https://ised-isde.canada.ca/site/innovation-better-canada/en/consumer-privacy-protection-act.
[3] *Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications).* . URL https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32002L0058.
[4] *Directive 2009/136/EC of the European Parliament and of the Council of 25 November 2009 amending Directive 2002/58/EC concerning the processing of personal data and the protection of privacy in the electronic communications sector (Electronic Communications Data Protection Directive).* . URL https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32009L0136.
[5] *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).*
[6] Alphabet inc. - annual report pursuant to section 13 or 15(d) of the securities exchange act of 1934. URL https://abc.xyz/assets/investor/static/pdf/20230203_alphabet_10K.pdf.
[7] In-app advertising spending worldwide from 2018 to 2028. URL https://www.statista.com/forecasts/1416806/in-app-advertising-spending-worldwide.
[8] Data protection framework list, . URL https://www.dataprivacyframework.gov/list.
[9] Privacy - control - apple. Online: https://www.apple.com/privacy/control/, . accessed 2024-01-27.
[10] checkra1n. Online: https://checkra.in/, . accessed 2023-10-10.
[11] Test your implementation on the platform. Online: https://developers.facebook.com/docs/audience-network/setting-up/testing/platform, . accessed 2023-09-29.
[12] Debug events | google analytics for firebase. Online: https://firebase.google.com/docs/analytics/debugview#android, . accessed 2024-02-28.
[13] Get Started | Android | Google for Developers. Online: https://developers.google.com/admob/android/quick-start, . accessed 2023-10-18.
[14] Get Started | iOS | Google for Developers. Online: https://developers.google.com/admob/ios/quick-start, . accessed 2023-10-18.
[15] Amplitude sdks - ampltiude developers center. Online: https://www.docs.developers.amplitude.com/data/sdks/, . accessed 2023-10-18.
[16] How to setup european data residency | community. Online : https://community.amplitude.com/data-instrumentation-57/how-to-setup-european-data-residency-907, . accessed 2024-01-26.
[17] Api key not working | community. Online: https://community.amplitude.com/data-instrumentation-57/api-key-not-working-1958, . accessed 2024-01-26.
[18] Android Ad Network statistics and market share. Online: https://www.appbrain.com/stats/libraries/ad-networks, . accessed 2023-10-17.
[19] The Most Popular Ads & Monetization SDKs. Online: https://appfigures.com/top-sdks/ads/all, . accessed 2023-10-17.
[20] The Most Popular Analytics SDKs. Online: https://appfigures.com/top-sdks/analytics/all, . accessed 2023-10-17.
[21] Max mediation documentation. Online: https://dash.applovin.com/documentation/mediation/max/get-started-with-max, . accessed 2023-10-18.
[22] Getting started. Online: https://dev.appsflyer.com/hc/docs/getting-started, . accessed 2023-10-18.
[23] Exodus Privacy. Online: https://exodus-privacy.eu.org/, . accessed 2023-10-17.
[24] Firebase. Online: https://example.org, . accessed 2023-10-18.
[25] Flurry documentation - yahoo developer network. Online: https://developer.yahoo.com/flurry/docs/, . accessed 2023-10-18.
[26] Business Software and Services Reviews. Online: https://www.g2.com/categories/mobile-app-analytics/, . accessed 2023-10-17.
[27] Meta audience network. Online: https://developers.facebook.com/docs/audience-network/, . accessed 2023-10-18.
[28] Getting started - meta app events. Online: https://developers.facebook.com/docs/app-events/getting-started, . accessed 2023-10-18.
[29] Android - meta app events. Online: https://developers.facebook.com/docs/app-events/getting-started-app-events-android, . accessed 2023-10-18.
[30] Most popular installed ad network software development kits (SDKs) across Android apps worldwide as of July 2023. Online: https://www.statista.com/statistics/1035623/leading-mobile-app-ad-network-sdks-android/, . accessed 2023-10-17.
[31] Most popular installed analytics software development kits (SDKs) across Android apps worldwide as of July 2023. Online: https://www.statista.com/statistics/1035612/leading-mobile-app-analytics-sdks-android/, . accessed 2023-10-17.
[32] Welcome to unity ads. Online: https://docs.unity.com/ads/en-us/manual/UnityAdsHome, . accessed 2023-10-18.
[33] Vungle SDK help center. Online : https://support.vungle.com/hc/en-us/categories/200269670-Vungle-SDK, . accessed 2024-02-22.
[34] Amplitude. Sdk quickstart guide | eu data residency. Online : https://www.docs.developers.amplitude.com/data/sdks/sdk-quickstart/#eu-data-residency_1. accessed 2024-01-26.
[35] Rebecca Balebako, Abigail Marsh, Jianliu Lin, Jason Hong, and Lorrie Faith Cranor. The privacy and security behaviors of smartphone app developers. In *The Symposium on Usable Security and Privacy*, 2014.
[36] Nataliia Bielova, Laura Litvine, Anysia Nguyen, Mariam CHammat, Vincent Toubiana, and Estelle Hary. The effect of design patterns on (present and future) cookie consent decisions. In *USENIX Security Symposium*, 2024.
[37] Reuben Binns, Ulrik Lyngs, Max Van Kleek, Jun Zhao, Timothy Libert, and Nigel Shadbolt. Third party tracking in the mobile ecosystem. In *Proceedings of the ACM Conference on Web Science*, 2018.
[38] Soumaya Boussaha, Lukas Hock, Miguel Bermejo, Ruben Cuevas Rumin, Angel Cuevas Rumin, David Klein, Martin Johns, Luca Compagna, Daniele Antonioli, and Thomas Barber. Fp-tracer: Fine-grained browser fingerprinting detection via taint-tracking and multi-level entropy-based thresholds. In *Proceedings on Privacy Enhancing Technologies*, 2024.
[39] Alban Diquet. SSL Kill Switch 2. Online: https://github.com/nabla-c0d3/ssl-kill-switch2. accessed 2023-06-29.
[40] David Doty. A Reality Check On Advertising Relevancey And Personalization. *Forbes*, August 2019.
[41] Eugen Ehemann and Martin Selmayr, editors. *General Data Protection Regulation*. 3rd edition, 2024.
[42] Anirudh Ekambaranathan, Jun Zhao, and Max Van Kleek. Understanding value and design choices made by android family app developers. In *The ACM CHI Conference on Human Factors in Computing Systems*, 2020.
[43] Federal Constitutional Court of Germany. BVerfGE 65, 1, 15. December 1983.
[44] Sibylle Gierschmann. Telekommunikation-telemedien-datenschutz-gesetz und die dsgvo. In *Multimedia und Recht*, 2023.
[45] Catherine Han, Irwin Reyes, Alvaro Feal, Joel Readon, Primal Wijesekera, Vallina-Rodriguez, Elazar Amit, Kenneth Bamberger, and Serge Egelman. Do you get what you pay for? comparing the privacy behaviors of free vs. paid apps. In *The Workshop on Technology and Consumer Protection*, 2019.
[46] Catherine Han, Irwin Reyes, Alvaro Feal, Joel Readon, Primal Wijesekera, Vallina-Rodriguez, Elazar Amit, Kenneth Bamberger, and Serge Egelman. The price is (not) right: Comparing privacy in free and paid apps. In *Proceedings on Privacy Enhancing Technologies*, 2020.
[47] International Association of Privacy Professionals (iapp). Schrems I. Online: https://iapp.org/resources/article/schrems-i/. accessed 2023-09-01.
[48] Simon Koch, Benjamin Altpeter, and Martin Johns. Github: The ok is not enough. Online: https://github.com/the-ok-is-not-enough/scala-plotalyzer/tree/master/src/main/scala/de/tubs/cs/ias/plotalyzer/trackerAnalysis. accessed 2023-10-10.
[49] Simon Koch, Malte Wessels, Benjamin Altpeter, Madita Olvermann, and Martin Johns. Keeping Privacy Labels Honest. In *Proceedings on Privacy Enhancing Technologies*, 2022.
[50] Simon Koch, Benjamin Altpeter, and Martin Johns. The OK Is Not Enough: A Large Scale Study of Consent Dialogs in Smartphone Applications. In *USENIX Security Symposium*, 2023.

[1] https://languagetool.org/
[2] https://app.grammarly.com/

[51] Konrad Kollnig, Reuben Binns, Pierre Dewitte, Max v. Kleek, Ge Wang, Daniel Omeiza, Helena Webb, and Nigel Shadboldt. A fait accompli? an empirical study into the absence of consent to third-party tracking in android apps. In *USENIX Symposium on Usable Privacy and Security*, 2021.
[52] Konrad Kollnig, Anastasia Shuba, Reuben Binns, Max Van Kleek, and Nigel Shadbolt. Are iPhones Really Better for Privacy? A Comparative Study of iOS and Android Apps. In *Proceedings on Privacy Enhancing Technologies*, 2022.
[53] Tianshi Li, Yuvraj Agarwal, and Jason I. Hong. Coconut: An ide plugin for developing privacy-friendly apps. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2018.
[54] Tianshi Li, Elizabeth Louie, Laura Dabbish, and Jason I. Hong. How developers talk about personal data and what it means for user privacy: A case study of a developer forum on reddit. In *The ACM CHI Conference on Human Factors in Computing Systems*, 2020.
[55] Celine Matte, Nataliia Bielova, and Cristiana Santos. Do cookie banners respect my choice? : Measuring legal compliance of banners from iab europe's transparency and consent framework. In *IEEE Symposium on Security and Privacy*, 2020.
[56] Abraham H. Mhaidli, Yixin Zou, and Florian Schaub. "we can't live without them!" app developers' adoption of ad networks and their considerations of consumer risks. In *USENIX Symposium on Usable Privacy and Security*, 2019.
[57] Mitmproxy Project. mitmproxy - an interactive HTTPS proxy. Online: https://mitmproxy.org/. accessed 2023-06-29.
[58] Trung Tin Nguyen, Michael Backes, Ninja Marnau, and Ben Stock. Share First, Ask Later (or Never?) Studying Violations of {GDPR's} Explicit Consent in Android Apps. In *USENIX Security Symposium*, 2021.
[59] Trung Tin Nguyen, Michael Backes, and Ben Stock. Freely given consent? studying consent notice of third-party tracking and its violations of gdpr in android apps. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2022.
[60] noyb. European Commission Gives EU-US Data Transfers Third Round CJEU. Online: https://noyb.eu/en/european-commission-gives-eu-us-data-transfers-third-round-cjeu. accessed 2023-09-01.
[61] European Court of Justice. Case c-210/16, unabhängiges landeszentrum für datenschutz schleswig-holstein v. wirtschaftsakademie schleswig-holstein gmbh, 2018-05-06. Available: ECLI:EU:C:2018:388.
[62] European Court of Justice. Case c-362/14, maximillian schrems v. data protection commissioner, european court of justice, 2020-07-16. Available: ECLI:EU:C:2015:650.
[63] European Court of Justice. Case c-311/18, data protection commissioner v. facebook ireland ltd and maximillian schrems, 2020-07-16. Available: ECLI:EU:C:2020:559.
[64] OneTrust. The Definitive Guide to Schrems II. Online: https://www.dataguidance.com/resource/definitive-guide-schrems-ii. accessed 2023-09-01.
[65] Articke 29 Data Protection Working Party. Opinion 9/2014 on the application of directive 2002/58/ec to device fingerprinting. Adopted on 25 November 2014.
[66] Amogh Pradeep, Muhammad Talha Paracha, Protick Bhowmick, Ali Davanian, Abbas Razaghpanah, Taejoong Chung, Martina Lindorfer, Narseo Vallina-Rodriguez, Dave Levin, and David Choffnes. A comparative analysis of certificate pinning in android & ios. In *Proceedings of the ACM Internet Measurement Conference*, 2022.
[67] Ole André V. Ravnås. Frida. Online: https://frida.re/docs/android/. accessed 2023-06-29.
[68] Abbas Razaghpanah, Rishab Nithyanand, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Mark Allman, Christian Kreibich, and Phillipa Gill. Apps, Trackers, Privacy, and Regulators: A Global Study of the Mobile Tracking Ecosystem. In *The Network and Distributed System Security Symposium*, 2018.
[69] Irwin Reyes, Primal Wijesekera, Joel Reardon, Amit Elazari Bar On, Abbas Razaghpanah, Narseo Vallina-Rodriguez, and Serge Egelman. "Won't Somebody Think of the Children?" Examining COPPA Compliance at Scale. In *Proceedings on Privacy Enhancing Technologies*, 2018.
[70] SensePost. Objection - Runtime Mobile Exploration. Online: https://github.com/sensepost/objection. accessed 2023-06-29.
[71] Kate Shilton and Daniel Greene. Linking platforms, practices, and developer ethics: Levers for privacy discourse in mobile application development. In *Journal of Business Ethics*, 2019.
[72] Sooel Son, Daehyeok Kim, and Vitaly Shmatikov. "what mobile ads know about mobile users". In *The Network and Distributed System Security Symposium*, 2016.
[73] Mohammad Tahaei and Kami Vaniea. "developers are responsible": What ad networks tell developers about privacy. In *The ACM CHI Conference on Human Factors in Computing Systems*, 2021.
[74] Mohammad Tahaei, Alisa Frik, and Kami Vaniea. Deciding on personalized ads: Nudging developers about user privacy. In *USENIX Symposium on Usable Privacy and Security*, 2021.
[75] Mohammad Tahaei, Kopo M. Ramokapane, Tinashi Li, Jason I. Hong, and Awais Rashid. Charting app developers' journey through privacy regulation features in ad networks. In *Proceedings on Privacy Enhancing Technologies*, 2022.
[76] Nicolas Viennot, Edward Garcia, and Jason Nieh. A measurement study of google play. In *The ACM international conference on Measurement and modeling of computer systems*, 2014.

Appendices

Table 7a and Table 7b display the position each SDK received by the considered ranking website. SDKs that were not placed in the top ten of a ranking receive position 11. We aggregate the ranks via average to determine our combined rank.

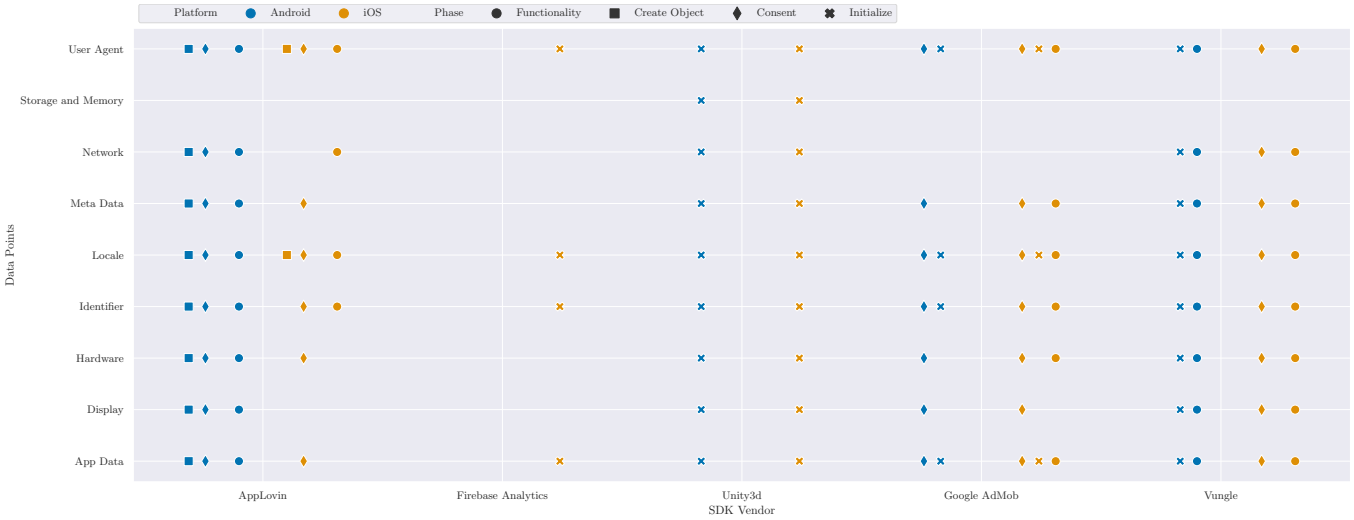

Figure 4: Transmitted data across the different phases without consent for SDKs providing a proper functionality to manage user consent.

Rank	Name	AppBrain	Appfigures	Exodus	G2	Statista	Ø Score
1	AdMob	1	1	1	3	1	1.4
2	Facebook Ads	2	2	2	1	2	1.8
3	Unity3D Ads	4	3	3	2	3	3
4	AppLovin	5	4	4	7	6	5.2
5	IAB Open Measurement	3	5	5	11	4	5.6
6	AdColony	9	8	6	11	7	8.2
7	Vungle	7	7	7	11	11	8.6
8	StartApp	11	6	11	11	5	8.8
9	IronSource	6	7	11	11	11	9.2
10	Google Ad Manager	11	11	11	4	11	9.6
11	AdLib	11	5	11	11	11	9.8
12	AdGate Media	11	11	11	5	11	9.8
13	Chartboost	11	6	11	11	11	10
14	Storyly	11	11	11	6	11	10
15	Flurry	11	11	8	11	11	10.4
16	AppsFlyer	8	11	11	11	11	10.4
17	Appodeal	11	11	11	8	11	10.4
18	Moat	11	11	9	11	11	10.6
19	InMobi	9	11	11	11	11	10.6
20	Supersonic	11	9	11	11	11	10.6
21	Chocolate	11	11	11	9	11	10.6
22	Huawei Mobile Services	11	11	10	11	11	10.8
23	AdColony	10	11	11	11	11	10.8
24	Appnext	11	10	11	11	11	10.8
25	ONE by AOL	11	11	11	10	11	10.8

(a) Advertisment

Rank	Name	AppBrain	Appfigures	Exodus	G2	Statista	Ø Score
1	Google Analytics	2	4	3	1	2	2.4
2	Firebase	1	1	1	11	1	3
3	Meta App Events	11	3	2	11	3	6
4	AppsFlyer	4	11	6	6	6	6.6
5	Amplitude	11	6	11	2	6	7.2
6	Flurry	9	5	8	11	4	7.4
7	Fabric	11	2	11	11	2	7.4
8	IronSource	3	11	7	11	11	8.6
9	Mixpanel	11	8	11	3	11	8.8
10	IAB Open Measurement	11	11	11	11	3	9.4
11	Clever Tap	11	7	11	7	11	9.4
12	Adjust	5	11	10	11	11	9.6
13	Moat Analytics	6	11	9	11	11	9.6
14	Google Tag Manager	11	11	4	11	11	9.6
15	GlassBox	11	11	11	4	11	9.6
16	Yandex Metrics	8	11	11	11	8	9.8
17	App Center Analytics	11	5	11	11	11	9.8
18	AppLovin	11	11	5	11	11	9.8
19	Adobe Analytics	11	11	11	5	11	9.8
20	AppCenter	7	11	11	11	11	10.2
21	Segment	11	7	11	11	11	10.2
22	GameAnalytics	10	9	11	11	11	10.4
23	Web Engage	11	8	11	11	11	10.4
24	Google Analytics 360	11	11	11	8	11	10.4
25	Pendo	11	11	11	9	11	10.6

(b) Analytics

Table 7: Combined ranks of the top ten SDKs for each considered ranking website. If an SDK was not contained in the Top Ten, it received a rank of 11.