University of Limoges

A study on Consent Management Platform in Android Applications

Student: Thi-Mai-Phuong NGUYEN
Supervisor: Assoc. Prof. Mathieu CUNCHE

This work was done at the Inria Privatics team hosted in the CITI-lab of INSA-Lyon.

August, 2022

Acknowledgments

This report would not be complete without the professors and friends who have always supported and motivated me during the whole time. First of all, I would like to express my most heartfelt and appreciative thanks to my supervisor, Mr. Mathieu Cunche. From the first days of conversing about the internship, he always listened to my inexperienced ideas and gave me a lot of useful advice, pointed out potential directions for them. Not only that, he was always given attention to my spiritual life and facilitated me to overcome those invisible anxieties. We've only worked together for 5 months, but he has motivated me to continue on the route to understanding as well as protecting privacy in this fast-paced technology era.

I must also thank Mr. Duong Hieu Phan and the Vingroup scholarship program for funding my two years of master's study in France. It is not only a scholarship that allows me to continue my higher education, but the program also gave me more motivation to make my dream a reality. Once again, thank you for your contributions to Vietnam and for giving wings to the dreams of our young generation.

My next thanks go to my lecturers at the CRYPTIS master's program of the Faculty of Science and Technology, University of Limoges. All of your classes that I have attended over the years have opened up new a whole new frontier for me. Although I faced a lot of barriers from the beginning, especially the language difference, with the help of professors I was able to survive and then grow up gradually. I would like to give special thanks to Mr. Emmanuel Conchon, who always listened, supported, and encouraged me not only during my studies, and exams but also during my internship job search. Without that enthusiastic help, I surely would not be here today. Hopefully, more and more international students will have the opportunity to follow the CRYPTIS master's program and thereby find their passion.

Sincere thank to Ms. Hoa Nguyen, my college friend and also the person who introduced me to the Vingroup scholarship. Without her sharing experiences and encouragement, I would not have had the opportunity to do great things like now. Thanks to all of my Vietnamese friends in Limoges: Tintin, Thao Doan, Dee, Tux, Danh Nam, Duy, Duc Anh, Lam, and Nhat Bui for always being with me from the first day I came to France. Thank you to Mr. & Mrs. Hoan-Linh for the most delicious milk tea cups and finally, to my dear friend Nhi Pham who is always ready to do silly things with me. In addition, I would like to thank Ms. Chi Vu, Mr. Tuyen Nguyen, and other ex-coworkers for helping me a lot in implementing some of the experiments in this report. I am also thankful to all my friends at the CITI lab for helping me open up more with enjoyable talks every day: Souhaiel BenSalem, Linda Soumari, etc.

Last but not least, my appreciation extends to my beloved family who always supports my decisions unconditionally. Thank you for allowing me to spread my wings and pursue my dreams.

Abstract

Privacy has always been a serious issue that needs attention for many years. In the past, privacy has always focused on personal information and standard data that can be used to cover, profiling, discriminate, impersonate others and commit crimes. However, since the internet has grown and technology has progressed further, the issue of privacy has been developed to the next level with newer visions. Currently, people's daily activities, as well as careers, begin to revolve around the term "online". Following that, business companies started the ideas of internet marketing, personalized advertising, etc. with their often-stated purpose of "enhancing the user experience". Additionally, many companies have started to convert the collected user data into a source of profit by selling them to other companies. As so extensively personal data has been collected and analyzed through the internet, governments have been forced to step in, creating strict regulations on the privacy of internet users. These regulations are designed to give individuals ultimate control over how their data is used. Hence, obtaining user consent has become a mandatory step before personal data is collected and shared with third parties.

While consent management mechanisms in websites have been thoroughly studied, very little attention has been spent on mobile applications. In this report, we give out the basics of consent management in mobile apps and its possible approach implementations for developers. In addition, as a result of performing some experiments, we present some of the consent management platform providers being used by mobile applications today. Along with that, indications to recognize their presence in the application will also be given. Variously, we will describe the steps as well as the results of some experiments which we performed to learn about consent mechanisms in mobile apps. And with all results that we got through this study process, a judgment about the possible privacy risks associated with consent management platforms can be conducted. Finally, we will point out potential future research directions related to consent management platforms in mobile apps.

Contents

List of Figures

List of Tables

Listings

INTRODUCTION

1.1 Context and related works

Over the past decade, with the revolution of internet connectivity as well as the innovation of technology, data about every aspect of our activities, behaviors, and lifestyles have been collected and then become "articles" for exchange or purchase in the digital economy. However, in reality, there are still many people who have been underestimating the importance of online privacy without knowing that the amount of information they are sharing is not only on social networks but also through using a browser or mobile application [33]. Recognizing the seriousness of the matter, numerous privacy regulations have been put in place so that we all have better control over our personal data such as European Union's General Data Protection Requirement (GDPR) [1], the California Consumer Privacy Act (CCPA) [3], the China Personal Information Protection Law (PIPL) [5], etc. Principal among them are regulations that aim to establish for individuals the right to choose what part of their data can be accessed by others and also can control the manner, scope, and timing of its exposure. Thus far, these regulations have made progress in the current situation of collecting and analyzing personal data on the internet, but more progress is necessary [18, 2, 31, 19, 54, 32, 9].

Amended in 2019, and came into force in May 2011, the Electronic Privacy Directive (EPD) [48], concerns the processing of personal data and the protection of privacy in the electronic communications sector. Regardless, it has become known as the "cookie law" since its most significant impact was the increase in cookie consent pop-ups on websites after it was passed. Following that impact, there has been a lot of research [34, 35, 7, 23, 27, 41, 21, 45, 46] done on the consent management platform of the websites, giving inquisitive readers a more detailed and complete picture of its working mechanism and potential violations hidden inside. As can be seen, the researchers pay so much attention to cookie consent pop-ups on the web platform. Meanwhile, most of them forget that these privacy regulations also apply to mobile ecosystems.

According to a statistics article from Datareportal in July 2022 [8], there are 5.34 billion unique mobile users in the world, equivalent to 66.9% of the population. With such a large number of users, it's no surprise that the mobile application industry is booming. Along with that is a lot of privacy-related issues in mobile applications that not all mobile users are aware of. Several studies have performed privacy compliance analyses of a small number of apps on the app store and recorded a list of violations [53, 55, 52]. And in 2019, Zimmeck et al. [47] decided to perform privacy compliance analyses on a set of 1 million Android apps from the Google Play Store and found broad evidence of potential non-compliance. In addition, Zimmeck et al. also pointed out that end users are often forced to accept third-party behavior related to privacy if they want to use the application's function. Indeed, in one of their studies, Ren et al. [24] observed 512 Android apps over 8 years of version history and concluded that an increased number of third-party domains receiving data directed to higher privacy risks over time. Recently, Nguyen et al. [36] revealed that 24,838 apps were sending personal data to third-party domains which relate to advertising without the user's prior consent. Meanwhile, Kollnig et al. [26] emphasize the lack of adequate mechanisms to collect the consent required under the current regulatory framework of mobile applications.

Confronted with these issues, a solution has been proposed - the consent management platform in mobile applications. Similar to the cookie consent pop-up of websites, a consent management platform is designed to obtain user consent regarding the use, sharing, or storage of personal data in mobile applications. However, until now, no study has provided an overview or analyzed the operation of this new mechanism. Because of that, there are still many questions on the consent management platform in mobile applications that need to be answered.

1.2 Research questions

In our work, we accomplish an investigation on a specific branch of applications that have the consent management platform. In more detail, our study aims to answer the following research questions:

- How does the consent management platform work?
- How can developers implement this mechanism into the application?
- With a consent management platform, are all mobile applications guaranteed to comply with user privacy regulations?
- Is consent linked with the device or account identifier?
- What factors do the global application rely on to determine the applicable regulatory framework?

In addition, several website consent management platform providers informed that they are providing consent management services for mobile applications also. Accordingly, we will also endeavor to answer other questions regarding consent management platform providers.

- What is the role of consent management platform providers in ensuring that mobile applications comply with privacy regulations?

1.3 Content & structure of the report

In the beginning of this report, we will present the background that the reader needs to know before starting to read the remains of the report (Section 2). Then, we present the basics of consent management in mobile apps (Section 3) and consent management implementation approaches (Section 4). We also provide a list of consent management platform providers that we have empirically identified (Section 5). Subsequently, to gain a more in-depth understanding of the consent management platform in mobile applications, we construct some experiments (Section 6) and present the obtained results (Section 7). Last but not least, our contributions along with future works and the summary will be discussed in the conclusion (Section 8).

1.4 Working environment

This report covers the research work which I have done during the 5 months internship at the Inria Privatics team hosted in the CITI-lab of INSA-Lyon under the supervision of Mr. Mathieu Cunche.

Inria [22] is France's national research institute for digital science and technology. Not only seeking world-class research or creating technological innovations, but Inria also accepts entrepreneurial risks. Currently, with 200 project teams, Inria has more than 3,900 researchers and engineers who are exploring new paths to meet ambitious challenges. Meanwhile, CITI [30] is an academic laboratory affiliated with INSA Lyon and INRIA. Research bringing together computer science, networking, and digital communications to solve challenging problems related to the development of the Internet is CITI Lab's main activity. With ten years of development, CITI with the expertise it holds has become a very original but challenging and almost unique laboratory in France.

Created in 2014, Privatics [49] is a research group under Inria and INSA-Lyon based in Grenoble and Lyon. The team's goal is to focus on protecting privacy in the digital world. Its activities are not only about research, learning about domains, and how the digital world evolves, but also about developing tools and systems to help enhance and protect privacy. The research of my supervisor, Mr. Mathieu Cunche, as a member of the Privatics team, also focuses on privacy issues related to information and communication technology. With his research experience in privacy specialization, he brought up this research topic about consent management platform with the final goal is characterize it in the Android ecosystem.

BACKGROUND

In this section, we provide some definitions of terms which relevant to our work. Additionally, we will also reiterate a few things regarding the legal background that users need to know before getting started with the rest of the report.

2.1 Definitions

Personal data

Different privacy regulations, such as GDPR or CCPA, will have diverse explanations of personal data, but in general, all of them compromise on the point that if the data can be used to identify - directly or indirectly by inference - a living individual, then it is personal data. In digital era, personal data can be e-mail address, IP address, connection logs, location data, etc.

Consent management platform

Following National Data Protection Commission (CNIL) definition [4], Consent Management Platforms allow the website or mobile app publishers to easily set up an interface to collect user consent. A pop-up will display when the user visits a website for the first time or at the first launch of a mobile application, showing different purposes of data collection and data controllers, and then storing the user's choices. Obviously, developers need to implement solutions that respect these choices.

Consent management platform providers

Today, many companies offer consent management platforms as a service, providing technologies that enable businesses to give customers control over their consent for sharing data through the consent banner. These companies are already known as consent management platform providers. According to public advertisements, consent management platform providers not only provide a consent management solution but also allow businesses to customize the consent banners that will appear. Along with that, they also allow businesses and developers to track or export user consent data through a management website. In addition, these providers assist businesses in scanning and listing all cookies and tracking technologies being used on websites or in mobile applications.

Third parties in mobile application

According to the dictionary [10], a third party is a person or group other than the two primarily involved in a situation, especially a dispute. Using third-party services in mobile app development is a common technique to incorporate more features and be more efficient in mobile apps. Application developers often choose them so that they don't have to invest extra time

and effort in redeveloping an existing module. However, third parties are often associated with privacy risks and developers are sometimes not fully aware of it. Besides third-party advertising services that are intentionally integrated by developers, there is a lot of third-party services that secretly collects user data and they are not aware of the developers. Actually, users are often unaware of the presence of these third parties when using an application. And whether intentionally or unintentionally, their data has also been collected implicitly.

Software Development Kit (SDK)

A software development kit (SDK) is a piece of software designed to integrate with any mobile application, allowing developers to build applications faster and in a more standardized way. Once integrated, third-party SDKs can have a significant impact on the performance, safety, security, and quality of applications.

Android Package Kit(APK)

Android Package Kit (APK) is a file format used by the Android operating system (or other Android-based operating systems) to install mobile applications or games. The APK file will include all the code and assets of the software program.

2.2 Legal background

In the scope of our report, the European Union's privacy laws are used as the main basis for legal analysis. They are basically made up of the Electronic Privacy Directive (ePD) and the General Data Protection Regulation (GDPR). These privacy laws regulate how data is allowed to be collected, processed, and stored. They empower individuals with certain rights, for example, the right to restrict the processing of their personal data, the right of access to their collected data, and the right to have personal data erased. Due to this, businesses that provide services to the European Union now have to comply with a data privacy landscape.

In the internet realm, the GDPR applies to any website and application, regardless of its origin, as long as it has users from the European Union. As an example, an application developed in Vietnam still needs to comply with GDPR and is required to have a legal basis for processing personal data if this application has users or provides services to the European Union. And according to article 6 [17] of GDPR, there are six legal bases for the lawful processing of personal data of data subjects in the European Union:

- the data subject has given consent;
- the processing is necessary for the performance of a contract;
- the processing is necessary for compliance with a legal obligation;
- the processing is necessary in order to protect the vital interests;
- the processing is necessary for the performance of a task carried out in the public interest;
- the processing is necessary for the purposes of the legitimate interests pursued by the controller or by a third party.

Thus, if do not have special legal bases, all applications and websites need the user's consent to collect, process, or store their data.

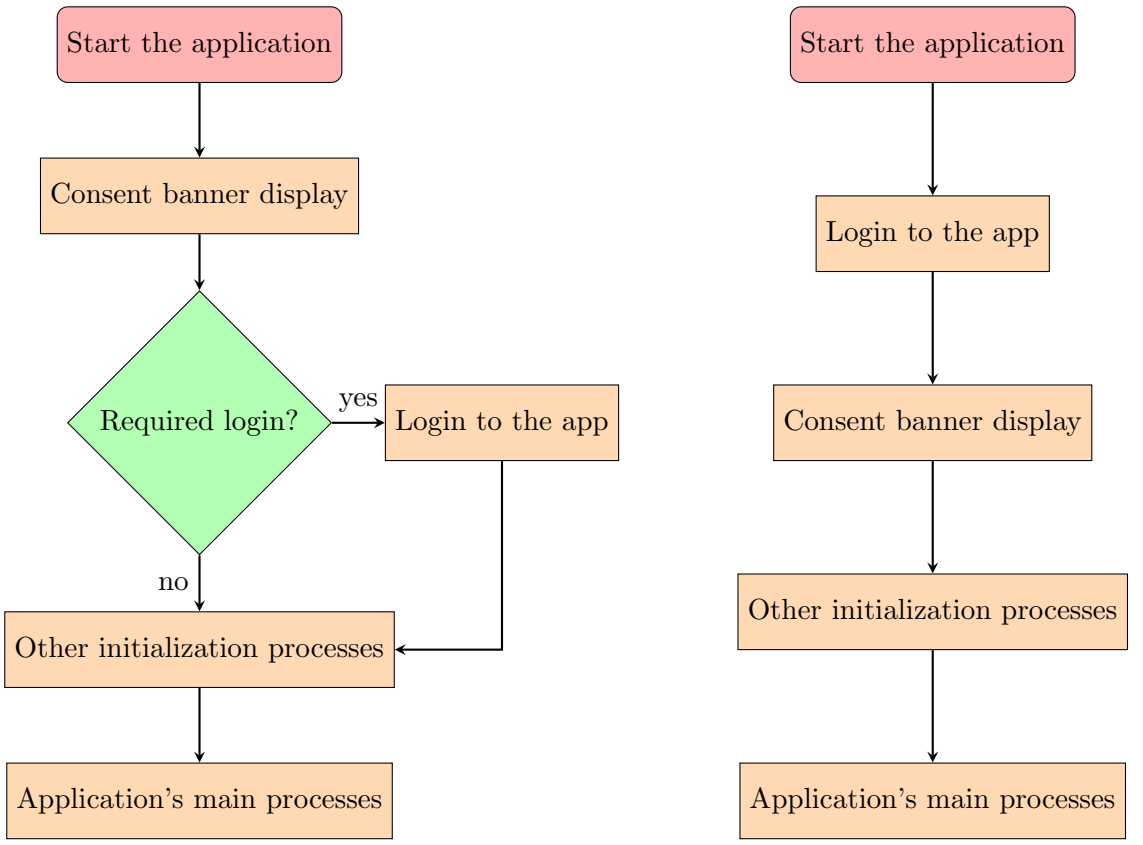
While the GDPR generally refers to rules governing the processing of personal data, the EPD has a more specific focus. The EPD specifically deals with data protection and freedom of communication in the field of electronic communications. This means that EPD is technology-neutral, covering not only websites and browser environments but also other types of technologies including applications on smartphones, tablets, smart TVs, and other devices. So if a web page requires a cookie banner today, a mobile app should also require a consent banner.

CONSENT MANAGEMENT ON MOBILE APPS

In the practice, the consent management process takes the responsibility of informing the user about what information is collected and giving the user permission to determine what personal data they are willing to share with a business. However, as normal users, we do not comprehend this internal process, and that is why we have several unknowing truths about it. In this section, we will present the basics of the consent management platform in the mobile app such as the user interface consent flow, the consent choices storage location, and the enforcement of the consent choices.

3.1 User interface consent flow

As we have already known, there are many different activities of an Android application like launching the app, logging in to an account, creating a new account, etc. And through a lot of testing applications, we observe the commonly time of appearance of consent banner.



(a) Consent banner displayed at startup

(b) Consent banner displayed after login

Figure 1: The flow of application with different time of appearance of consent banners

3.1.1 Consent banner displayed at startup

In most applications that do not require the login step, consent banners will be displayed as soon as the user starts the application for the first time. Besides, there are some cases where the application requires the user to log in to use the services, but the consent banner is still displayed first, before the login and the account registration step.

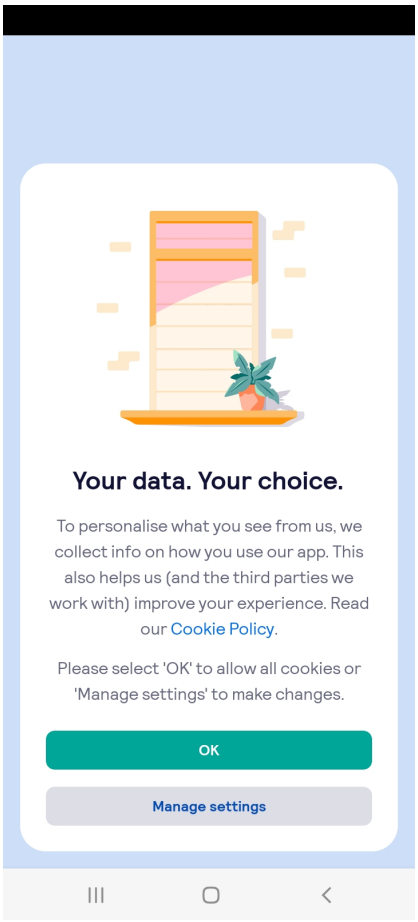


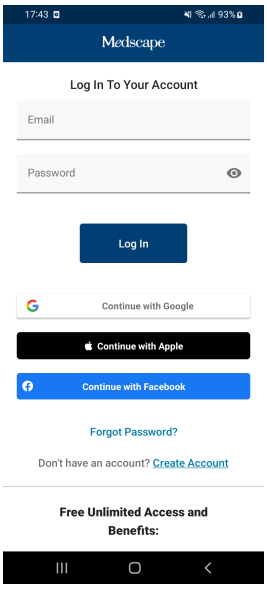
Figure 2: An app that displays the consent banner at the startup

Obviously, when the consent banner is displayed as soon as the user opens the app, the consent options will not be linked to the user account but will use other identifiers, such as device identifiers or maybe randomly generated identifiers.

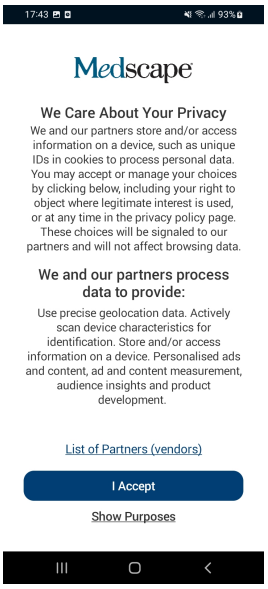
3.1.2 Consent banner displayed after login step

When signing up for a new account, users are asked to accept the privacy policy of the application. Anyway, in this subsection scope, we are just referring to the case where the consent banner still appears after the user completes the registration step. The other case will be mentioned in the next subsection 3.1.3.

If the consent banner appears after the login or account registration step, we expect the consent options to be linked to the user account. Following that relation, sharing consent across devices is what we can expect. Moreover, it is potential for sharing consent between the website and the mobile application also becomes possible. However, these are all assumptions, we need to conduct experiments to check this statement's validity.



(a) Application required login at first

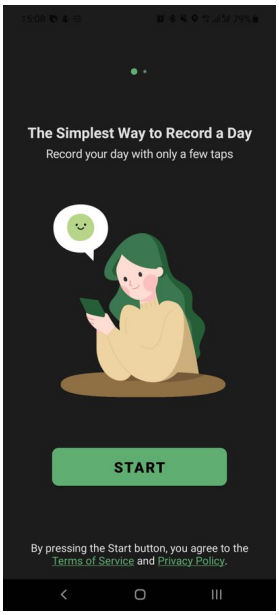


(b) Application displays consent banner

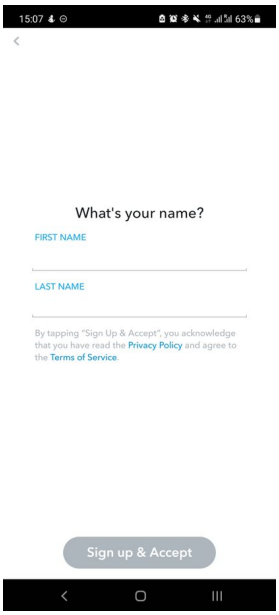
Figure 3: An app that displays the consent banner after login

3.1.3 Consent banner does not show up

In the case of applications that do not display the consent banner, the worst case is these applications have bypassed our statements when providing personal information to third parties or storing them for unclear purposes. However, there are also some cases where applications require the user's consent to store and share personal data as a mandatory step when starting to use the application or registering a new account on that application. In this case, the user is only notified that the application will use his personal information in the form of a written statement and they have no privilege to reject this.



(a) An app required acceptance at startup



(b) An app required acceptance at sign up

Figure 4: Examples of apps that force users to have implicit acceptance of privacy policy

3.2 Consent choice storage location

One of the next considerations in the consent management mechanism is the flow of information sharing. How are user consent choices and other information shared between the application and the servers of not only CMP service providers but also third parties?

According to the consent management platform providers, the user's choices will be recorded on the provider's server and thereby, allow developers to trace as well as export reports on user consent choices [42, 40]. Saving the consent choices on the online server is one of the important cornerstone steps to be able to use other services of the consent management platform providers such as synchronizing the user's consent across devices as well as across platforms [37, 13]. In another hand, following providers documentations [11, 44], the user's consent collected choices are also saved in the device's shared preferences. To verify the validity of these statements, we take an investigation into the SDK codes of one of those a provider - Quantcast.

We have been using the latest Quantcast SDK (`CMP-Choice-v2017_Android.aar`) which is published on April 7th, 2022. In the source code, we found the function which used to save consent choices to device's shared preferences: `saveAllConsentsPreferences()` (see full at Listing 3). After checking the value of the parameter is not null, it will be saved with a fixed key.

```
public final void saveAllConsentsPreferences(@NotNull String tcString,
    ↪ @NotNull String nonIabVendorConsents, /*..*/) {
    Intrinsics.checkNotNullParameter(tcString, "tcString");
    Intrinsics.checkNotNullParameter(nonIabVendorConsents, "
    ↪ nonIabVendorConsents");
    /*..*/
    setStringPreference(SharedStorageKeys.TC_STRING, tcString);
    setStringPreference(SharedStorageKeys.NON_IAB_VENDOR_CONSENT_HASH,
    ↪ nonIabVendorConsents);
    /*..*/
}
```

Listing 1: Consent choices are saved in the SharedPreferences (Quantcast)

On the other hand, we also found function `LiveData` that used to push user activities to the online server (see full at Listing 4). When the user taps on any button on the consent banner, that behavior will be sent to the provider server.

```
public final LiveData<String> consentAllAcceptation(@NotNull UIInteractions
    ↪ actionTag, @NotNull Regulation regulation) {
    Intrinsics.checkNotNullParameter(actionTag, "actionTag");
    Intrinsics.checkNotNullParameter(regulation, "regulation");
    String navigationTag = null;
    Boolean acceptAll = null;
    switch (WhenMappings.$EnumSwitchMapping$1[actionTag.ordinal()]) {
        case 1:
            acceptAll = Boolean.valueOf(true);
            navigationTag = String.valueOf(UIInteractions.ACCEPT_ALL);
            tracking.pushEvent(navigationTag, "click");
            return CoroutineLiveDataKt.liveData$default(null, 0L, new
    ↪ UI$consentAllAcceptation$1(acceptAll, regulation, null), 3, null);
        /*..*/
    }
    /*..*/
}
```

Listing 2: User behaviors are pushed to the online tracking tool (Quantcast)

From these two functions, we can give a conclusion that verifies the providers' statements: The consent choices are not only stored locally on the device, but also by the providers on their online servers.

3.3 Enforcement of the consent choices

On the other hand, when evaluating the correctness of the consent management platform in an application, we need to consider how the user consent choices are applied in the application. In other words, we should consider the flow of information exchanged between the application and the third parties. With the participation of consent management platform providers, the enforcement of consent choices will become more diverse during the application development process.

Nowadays, the majority of consent management platform providers are registered and compliant with the IAB Europe Transparency & Consent Framework (IAB TCF) [29]. Well-known names on the list [15] like Didomi, OneTrust, Quantcast, etc. not only commit to compliance but also use this framework to share user consent with third parties using Transparency & Consent string [14, 38, 43]. With default settings for the list of IAB vendors [16] available in their environment, the providers are committed to simplifying the consent management process for application developers. One of them is to limit problematic updates related to IAB vendors, especially when a new vendor is added to the list. Besides the vendors who support IAB TCF, there will still be many providers that do not. And in this case, each consent management platform provider has a different way of handling it. However, we have researched and recap a common workflow for enforcement of the consent choices as follows.

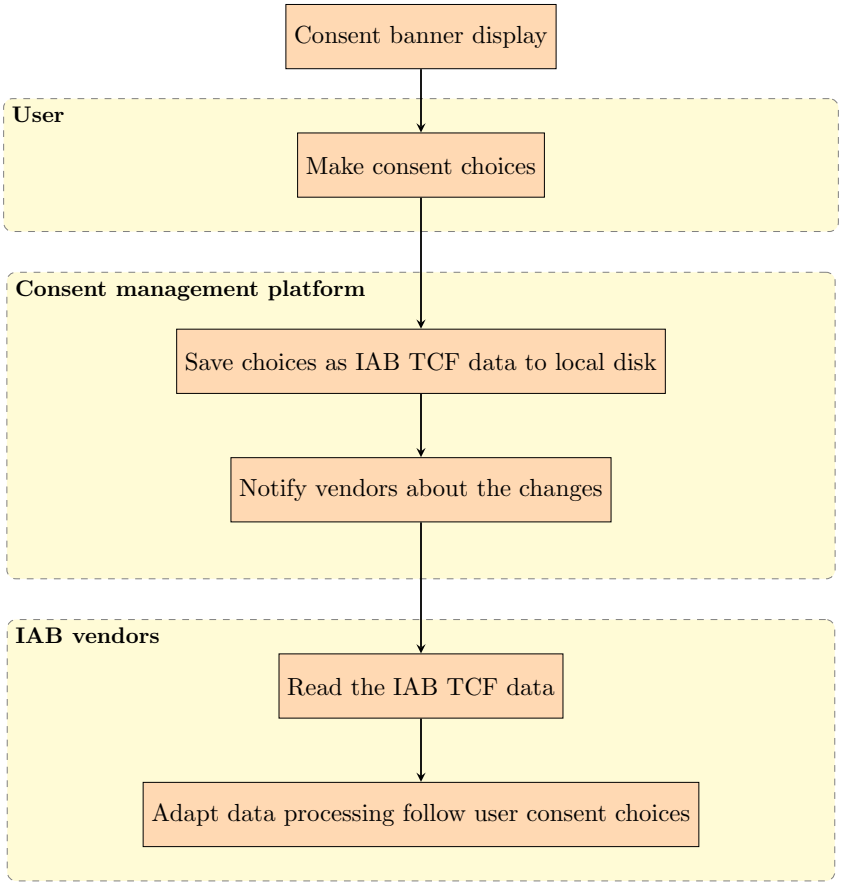


Figure 5: IAB TCF simplified workflow on the mobile platform

For third party vendors who support the IAB TCF, the developers can share the user consent with them through this framework. Similarly, we also have IAB CCPA Compliance Framework [28] that can be used to transfer the consent to the vendors who support this framework. A lot of research present IAB TCF workflow on the web platform [34, 46], but we have not found any studies specific to it on the mobile platform yet. To our observation, with these frameworks, providers can automatically adapt their data processing to respect user consent. The workflow can be simplified as shown in the figure 5.

For third party vendors that do not support any IAB specification, there are two possible cases as follows. Firstly, if vendors offer an API to notify them what the user consent status is, the application developers need to pass the user choices to their SDK. According to the consent, the vendors can update the process of data. At last, we have vendors without an API. More particularly, they do not offer any function that allows the developers to notify them about user consent updates. In this case, the developers must operate their initialization manually. Their SDK should not be loaded until the user gives consent. The workflow is presented in a simplified form as shown in figure 6.

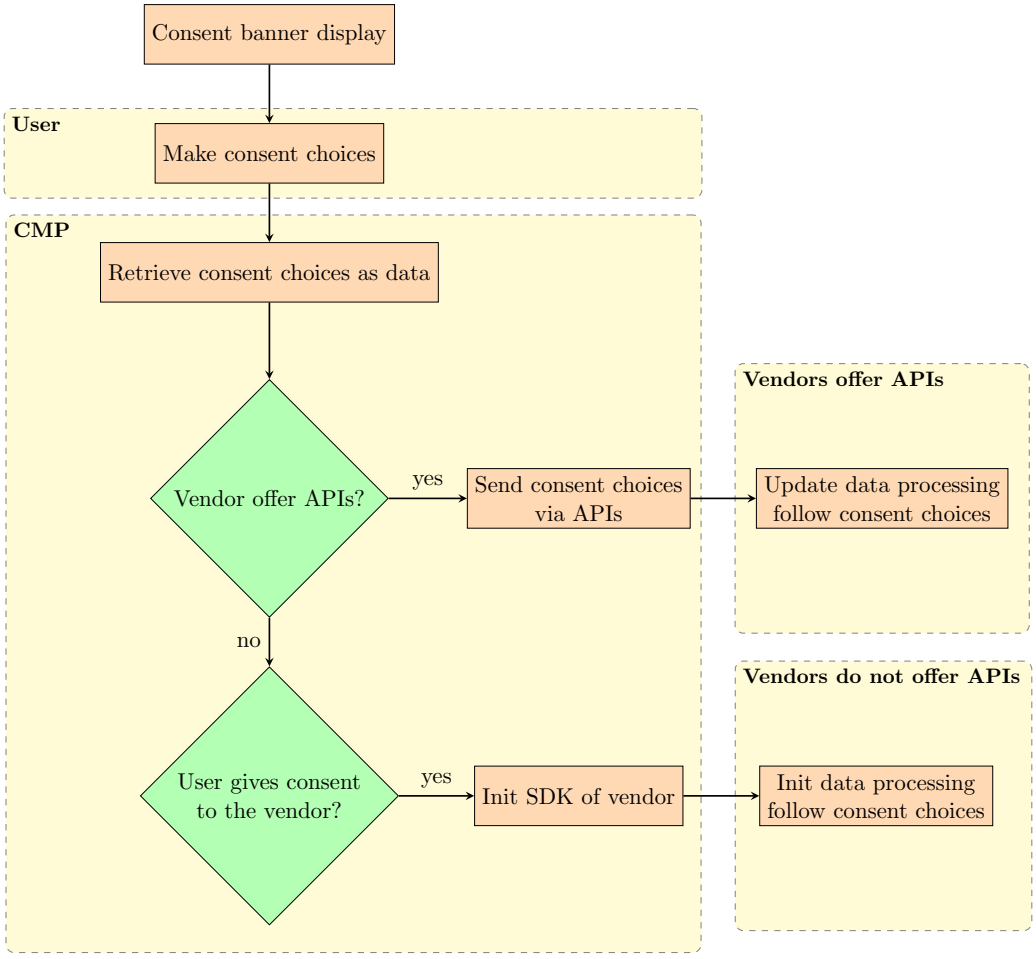


Figure 6: Simplified workflow to process consent for non-IAB TCF vendors.

We can summarize that user consent choices after being selected, will be processed by the consent management platform and directly sent to third-party SDKs. Obviously, CMP only plays a role in collecting and sharing user consent choices with third parties, not in restricting or controlling services that do not receive user consent. Finally, we still do not have a decisive answer on whether third parties are always compliant and deserve the trust of users or not.

3.4 The adjustment of consent management flow in global applications

According to an article [25] of Thorin Klosowski, the editor of privacy and security topics at Wirecutter, when Europe has a comprehensive privacy law - the General Data Protection Regulation (GDPR), the United States does not have a singular law that covers to all types of data. Instead, the United States owns a mix of laws like the Health Insurance Portability and Accountability Act (HIPPA), California Consumer Privacy Act (CCPA), Gramm–Leach–Bliley Act (GLBA), Children's Online Privacy Protection Rule (COPPA), etc. Each of those is designed to target only specific data types in special circumstances. Thus, it is apparent that we cannot deny the difference between the Europe privacy law and the United States laws. But how do the developers know which laws should be applied to the user of their application?

Following Google documentation [20], they based on IP address to detect the user's current country. And users are possible to see the content relevant to the general area which is also based on the user IP address. With this principle, we use VPN to change the current location of the IP address to the United States and then perform a small test with the application Booking.

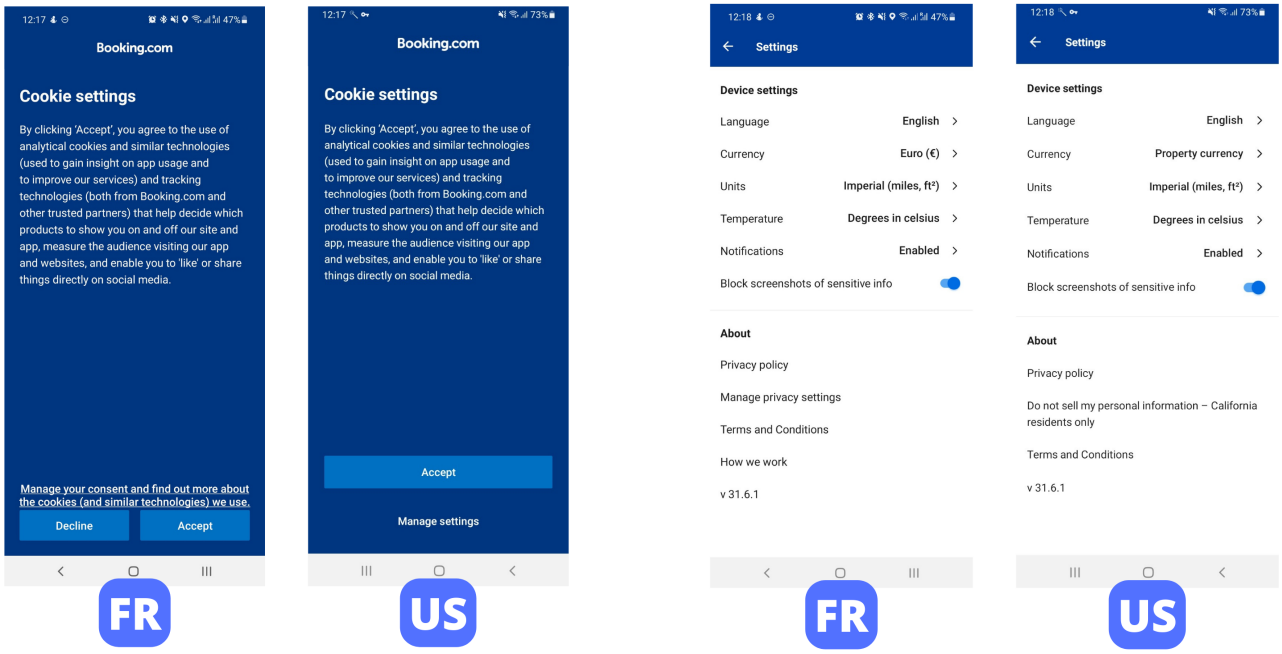


Figure 7: Compare consent setting on two versions of application Booking

As you can see in the figure 7, there are some differences between France and the United States versions of the application Booking. This small test has started our curiosity about the internal processes of global applications. The questions here are how do global application developers know the user's current location, the laws that apply to a user, and how can they ensure that their applications are compliant with mandatory laws? To answer these questions, we present some experiments intending to understand the internal processes of global applications which you can see in more details in section 6.3.

CONSENT MANAGEMENT IMPLEMENTATION APPROACHES

In this section, we would like to recap the ways that application developers can use to integrate the consent management platform (CMP) into the existing workflow. With the emergence of consent management platform providers, developers have more options when they can use the services that are accommodated by the providers. Besides, the self-development of a consent management platform for their application is also a viable option since the developed platform will be most compatible with the developer's requirements.

Through consent management platform provider documentation and also experiments on some applications, we explain some ways that application developers can use to integrate the consent management platform.

4.1 Use both SDK and UI of CMP providers

The simplest way to integrate the consent management platform is to use the available services from the providers. Currently, almost consent management platform providers offer a package service that includes a built-in mobile SDK and a justifiable user interface. Accordingly, the application developers only need to implement and use the available resources to follow the provider's instructions.

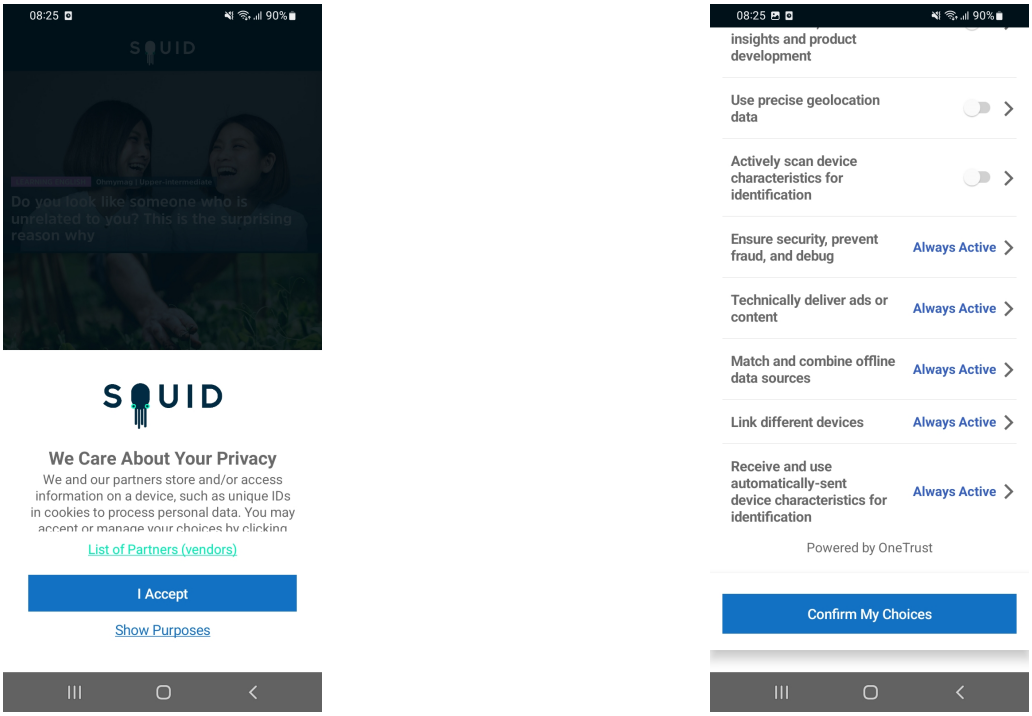


Figure 8: An app uses SDK and UI of OneTrust

With this integration, the consent banner will have the providers logo and we will also find the providers-specific package name when investigating the application's source code.

4.2 Only use CMP providers' SDK and build your own UI

Another option for developers is to embed the vendors' SDKs in the app and create their UI. Certainly, the logic and methods of the SDK remain the same, but developers need to create the layouts and logic of the interface from scratch. The link between the UI and the user consent management backend of SDK also needs to be created by the developers.

Although we have not found a lot of example applications yet, some providers announced they support this approach [12, 39, 50]. There's no denying that this approach gives developers the flexibility to develop specifically UI for their applications, but it will also take more time to create, maintain and update.

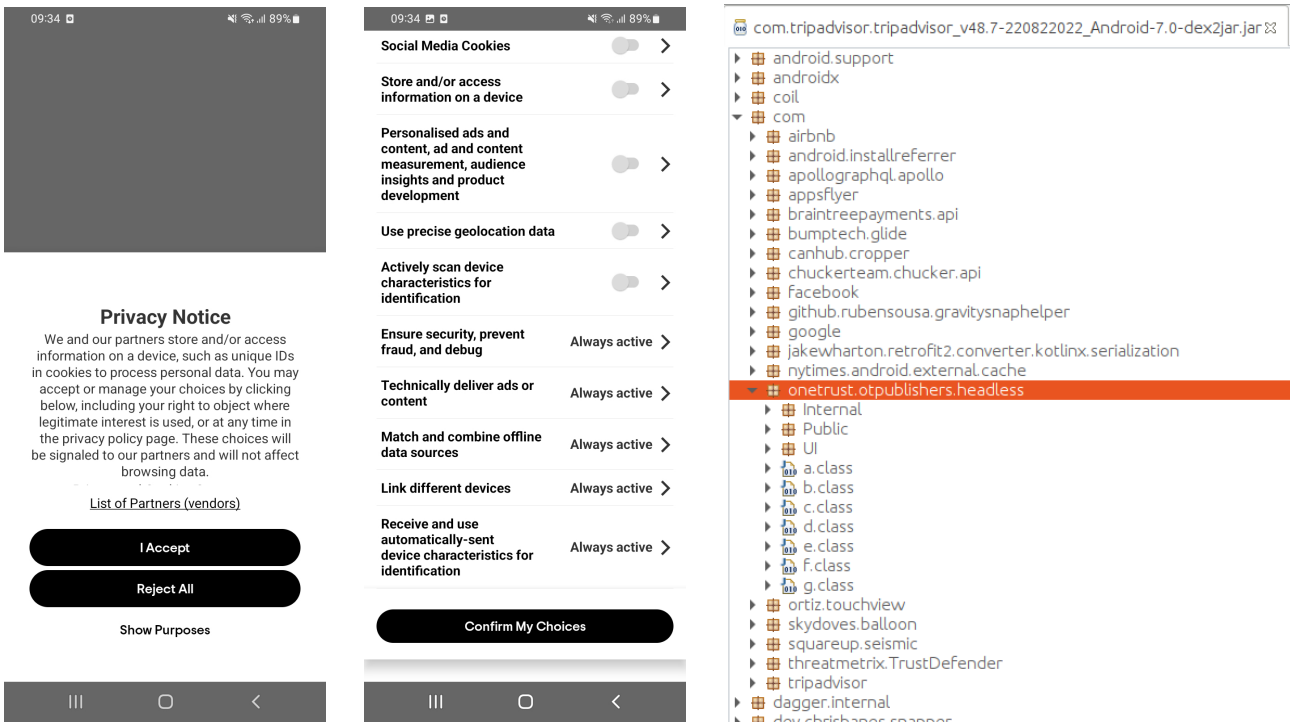


Figure 9: An app uses OneTrust SDK and build their own UI

As you can see in the figure 9, there is no appearance of the provider logo on the consent banner but in the source code of the application, we can easily detect their SDK package.

4.3 Only use CMP providers' API and build the remain parts

Although not many, there are still some consent management platform providers currently offer this option to application developers [39, 6, 51]. The most outstanding advantage of this option is to reduce the cost of services from the providers to the lowest. In addition, developers can develop management flow and create their own UI more flexibly. However, the trade-off for those advantages is many disadvantages that can predict from the beginning. Using these approaches, the developer must not only rebuild all the logic of the consent management platform but also create the logic and layout of the user interface from scratch. These tasks

will take more time than using the SDK of providers. As a matter of course, it also requires more experienced developers. Another disadvantage of this approach is that it becomes more difficult to update and change over time.

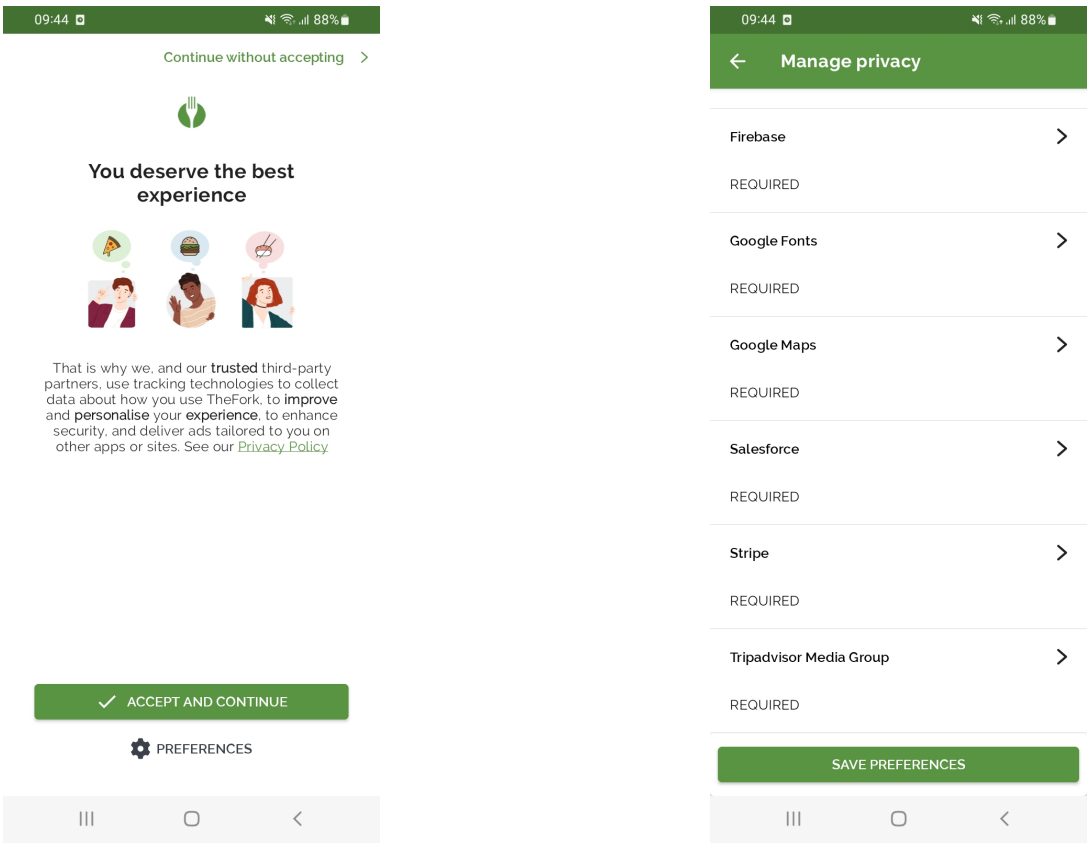


Figure 10: An app uses API of Crownpeak and build the remain parts

With this integration, the SDK package name of providers cannot be found when analyzing the application's source code, and the consent banner will also not show the provider's logo. Therefore, the only way to determine whether an application is using this type of service is to rely on network traffic between the application and the consent management platform provider's online server.

4.4 Self-developing consent management platform

Finally, we need to consider the option that developers build a consent management platform for their applications and do not use any additional services from providers. Obviously, with this approach, developers can maximize flexibility in building the management flow but we must also claim that this is the most time-consuming approach as having to rebuild everything needed from the beginning. In addition, if there is any hasty change from third parties or a small error in the consent management flow, it will be quite difficult to update the application as soon as possible. Unquestionably, this approach requires developers with many years of experience. They not only need to understand privacy policies but also need to understand how the app works to be able to create and maintain this self-developed consent management platform.

In our study, we classify applications that do not have any providers logo on banners, do not have SDK package of providers in the source code, and do not communicate to providers' servers are applications using the self-developed consent management platform.

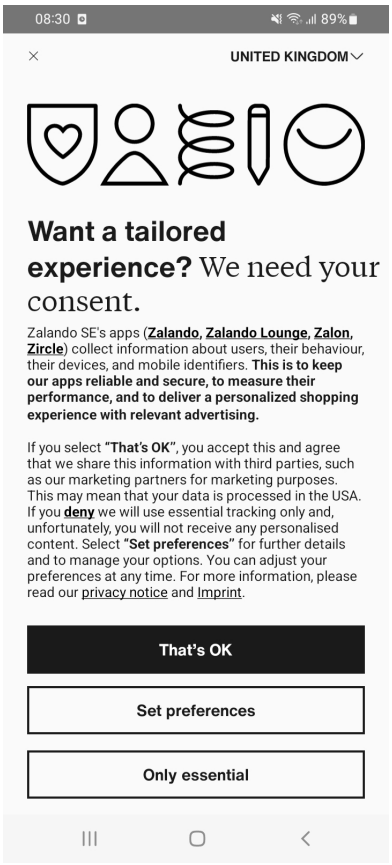


Figure 11: An app uses self-developing consent management platform

IDENTIFIED CONSENT MANAGEMENT PLATFORM PROVIDERS

In this section, we present the methods we used to detect the presence of consent management platform providers in applications. In addition, the list of identified providers as well as the indications of their presence are also given as the final result.

5.1 Methodology

Since the existence of Hybrid applications (they are deployed in a native container that uses a mobile WebView object) and the appearance of app wrapping tools, it become more difficult to get information about the SDKs used in any APKs. Therefore, besides the clear indication to detect the consent management platform providers like base on the package name, we also need other indications like network traffics or providers' logos on consent banners. The following are three methods we use with test applications. These methods will be applied from 1 to 3 respectively. If all three methods are completed but still cannot find out the existence of a provider, we will classify this application into the category of apps that are using the self-developing consent management platform.

Method 1: Decompile APK files

For the first step, we used some free tools such as dex2jar and jadx to decompile the APK files and JD-GUI to browse the reconstructed source code. We can base on the list of packages' names in the source code to categorize the applications into different groups that use the services of different consent management providers. Or in some cases, they use a self-developing consent management platform.

Method 2: Launch applications

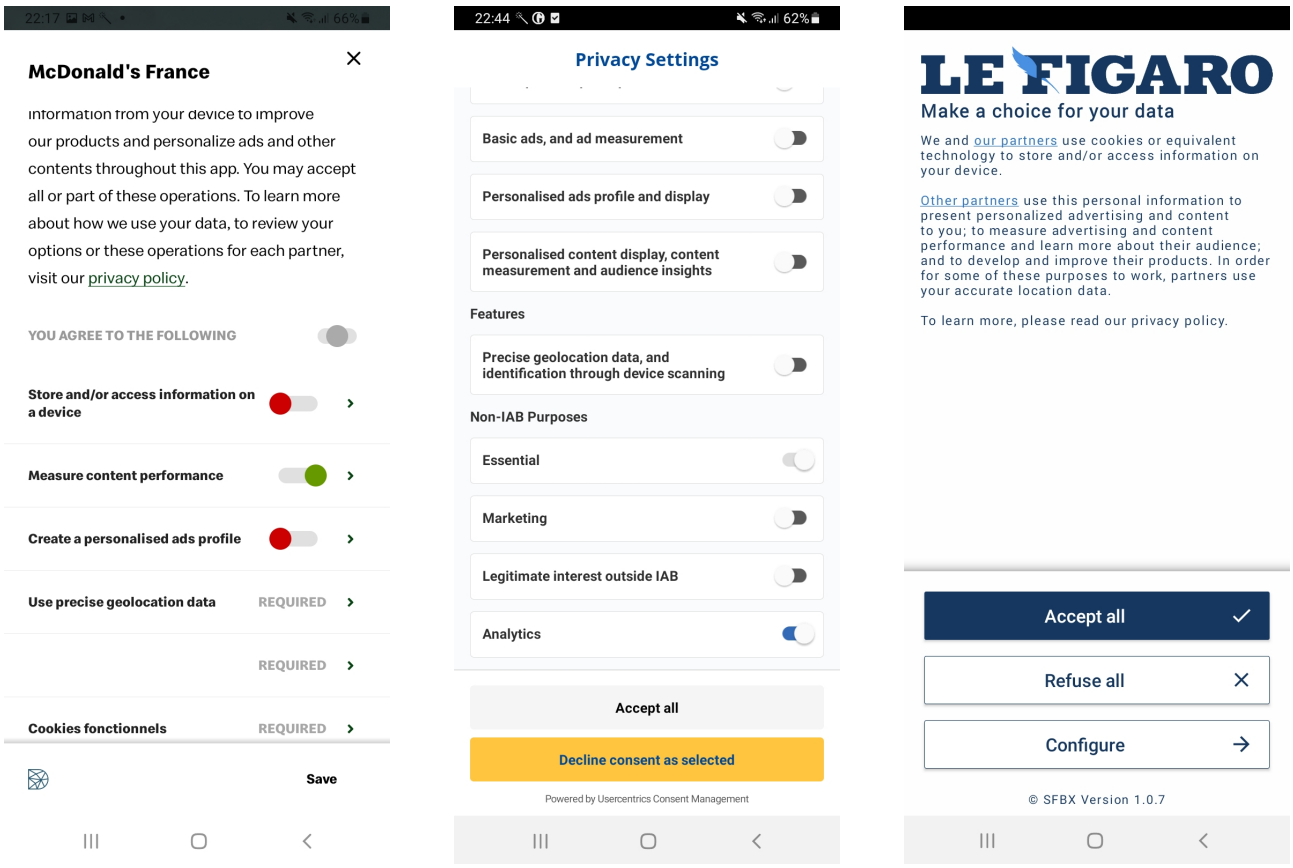
As we explained before, there are some special types of applications and we will not get any useful information by decompiling them. However, by running the application and examining their consent banners, we found some consent management platform providers will display their logos on these banners.

Method 3: Analyze network traffics

For applications that cannot be classified based on the above two methods, we perform an analysis of their Domain Name System (DNS) requests at the first launch. Evidently, when using the consent management platform provider's services, DNS requests are required to initiate a

connection to its servers. Therefore, this method will be the last but also surest method to find out the presence of consent management platform providers in applications.

5.2 Results



(a) An app uses service of Didomi (b) An app uses service of Usercentrics (c) An app uses service of SFBX

Figure 12: Some example of applications that use services of CMP providers.

A total of 8 consent management platform providers have been identified through a combination of static analysis of the APK and manual dynamic analysis of the interface (see 5.1 section). After performing experiments with more than two hundred random applications, we detect 30 applications that integrated consent management platform, including 24 applications using the providers' service. With these 24 applications, we summarize the following results.

5.2.1 Identified CMP Providers

We present 8 identified consent management platform providers for the mobile ecosystem in the following list.

1. OneTrust
2. Didomi.
3. Usercentrics
4. Sourcepoint

5. **Quantcast**
6. **SFBX**
7. **Crownpeak (Formerly Evidon)**
8. **TrustCommander**

Most of these providers support consent management platforms for both web and mobile ecosystems, promising more useful functionality for user consent management very shortly.

5.2.2 Indications to detect CMP providers

The following table includes indications that can be used to detect the consent management platform providers in applications. They are just some specific indications we get from the experiments, not all of them. To obtain even more accurate indications, a larger set of test applications and automated testing methods are needed.

Table 1: The list of indications which can be used to identify CMP Providers.

CMP Provider	Package name	Other indications
OneTrust	com.onetrust.otpublishers.headless	- Banners have "Powered by OneTrust" - Applications will make DNS request to mobile-data.onetrust.io
Didomi	io.didomi.sdk	- Banners have Didomi Logo - Applications will make DNS request to sdk.privacy-center.org
Sourcepoint	com.sourcepoint.ccpa_cmplibrary com.sourcepoint.gdpr_cmplibrary com.sourcepoint.cmplibrary	- Applications will make DNS request to cdn.privacy-mgmt.com
Usercentrics	com.usercentrics.sdk	- Banners have "Powered by Usercentrics Consent Management" - Applications will make DNS request to app.usercentrics.eu
Quantcast	com.quantcast.choicemobile	- Applications will make DNS request to quantcast.mgr.consensu.org
SFBX	com.sfbx.appconsent	- Banners have "SFBX Version x.x.xx" - Applications will make DNS request to collector.appconsent.io
Crownpeak	N/A	- Applications will make HTTP request to i.evidon.com/api/v1
TrustCommander	com.tagcommander.lib.privacy	- Applications will make HTTP request to cdn.trustcommander.net

METHODOLOGY OF EXPERIMENTS

Since there are many unanswered questions about consent management platforms in the mobile ecosystem, we try to perform as many as possible experiments to find out the answers or simply get a basic perspective on them and then identify potential directions for future research. On that account, we propose several experiments that can be divided into different groups by their objectives. In this report, we have three groups as follows:

- Group 1: Experiments used to detect the suspicious applications' actions were executed without users' explicit consent.
- Group 2: Experiments used to discover the storage location as well as identifiers used for consent settings.
- Group 3: Experiments used to explore the internal process of consent setting in global applications.

6.1 Detect the suspicious applications' actions

6.1.1 Experiment 1: Sniff network traffics

In the first experiment, we will capture all traffic of the testing application before the consent banner shows up. Therefore, we analyze the traffic to detect all third-party servers which this app communicates with. Based on that, we can point out the related privacy risks.

Tools used

To capture the network traffics of testing application and then save them in pcap format, we use a free application called PCAP Remote. For this experiment, we use the option "Capture traffics for a specific application" of PCAP Remote to ensure that there is no extraneous traffic flows that affect the results.

Subsequently, we use library Scapy in Python to analyze, and statistics the DNS traffics in collected pcap files. For this step, we begin with filtering out all the traffic packets which not have layer DNS. With the remaining packets, we list out all the domain names which can be found in the field "DNS/qd/qname" and perform the statistics on the number of occurrences of those domains.

Steps

1. Install the application
2. Open PCAP Remote, choose the testing application

3. Open the testing application
4. Close the application when the consent setting banner show up
5. Re-open PCAP Remote and save the captured network traffics as Pcap file.
6. Analyze, and statistics the DNS traffics in this Pcap file.

Outcome

A list of domains that test applications contact before consent selections are collected.

6.2 Discover the storage location as well as identifiers used for consent settings

The goal of the experiments in this group is to find out how user consent choices are stored. As discussed in the section 3.2, consent choices are not only stored in the device's shared preferences but also on the online servers of the consent management platform providers. However, we still do not know what identifiers the consent options will be attached to when stored on an online server. The experiment 2 (6.2.1) and experiment 3 (6.2.2) were created to better understand that. In another hand, the experiment 4 (6.2.3) was created to double-check the consent choices storage location.

6.2.1 Experiment 2: Switch account

The purpose of this experiment is to check if user consent choices are attached to the account identifier. In case the consent choices are preserved after the switch account action, then we can affirm that the consent choices are not attached to the account identifier.

Steps

1. We create a new account ACC_1 on testing application.
2. Login to the testing application by the ACC_1 and choose the consent choices for this account - CS_1.
3. Logout the account ACC_1 and restart application.
4. On the login screen, we create another account ACC_2 for testing application.
5. If the consent banner show up, finish the experiment. If nothing is display, go to step 6.
6. Check if the consent setting still be CS_1 or not.

Possible outcome

Is the consent setting changed when the application's account is changed? There are some possible outcomes:

- The consent choices are changed following the account change.
- The consent choices are not changed.
- The application shows up a fresh consent banner and requires the user to make the choices again.

6.2.2 Experiment 3: Change device

This experiment has the same objective as the previous experiment and is used to consolidate the results obtained earlier. If on the same account there are two different sets of consent choices on two different devices, we can affirm that the user consent choices are not attached to the account identifier.

Steps

1. We create a new account ACC_1 on the testing application on device A.
2. Login to the testing application on device A by the ACC_1 and choose the consent choice for this account - CS_1 and then close the testing application.
3. Login to the testing application on device B by the ACC_1. If the consent banner appears then we continue with step 4; if not, we compare the current consent choices of account ACC_1 on device A and device B if they are different or not.
4. On the consent banner, we choose the consent choices for account ACC_1 - CS_2 (different with CS_1) and then close the testing application.
5. On device A, open the testing application again (still logged as ACC_1);
6. Check if the consent choices still be CS_1 or changed to CS_2 (or changed to something else).

Possible outcome

Is the consent banner displayed at step 3 ?

- Consent banner displayed again on device B.
- Consent banner is not displayed on device B.

Is the consent setting changed following the device changing? We have some possible cases which can happen:

- The consent choices in the two devices are the same after step 4.
- The consent choices in the two devices are different.

6.2.3 Experiment 4: Reinstall application

We set up this experiment to check where user consent choices are stored. In case the application is uninstalled and then we reinstalled it again but the consent settings are kept as before, we can figure that the consent settings may be stored on an online server and tied to a device identifier or saved somewhere outside of the device's shared preferences.

Steps

1. Install the application, choose the consent choices as CS_1.

2. Uninstall the application.
3. Reinstall the application, check if the default consent choices are the same as CS_1 or not.

Possible outcome

Is the consent setting be kept the same as before after uninstall and reinstall actions? There are two possible cases:

- The consent setting is totally fresh as the first installation.
- The consent setting is the same as before uninstalled the app.

6.3 Explore the internal process of consent setting in global applications

As discussed in the section 3.4, there are some differences in the consent management flow when we change the IP address of the test device. However, changing the IP address can also cause many other related changes such as the country field of the Google account, the downloaded APKs file, etc. To better understand the determinants of this change, we present the following experiments.

6.3.1 Experiment 5: Build a test set of global apps

In the first step, we try to find global apps which exist on the Google Stores of different regions and countries. An application will be selected for further experiments if consent management flows on two versions of the application are not the same.

Steps

1. We disable the location services on both devices and we use the VPN to put device A in the United States and device B in France.
2. Still with the VPN on, we create a Google account on both devices, meaning that device A will have a Google account in the United States and device B a Google account in France (location of the account is automatically provided by Google based on the IP). This account is created through the devices.
3. Still using the VPN we install the app on both device A and device B. This means device A will retrieve the APK from the United States store while device B will retrieve it from the France store.
4. Finally, we start the app on both devices and observe the consent content and check if there is a difference.

Possible outcome

For the application, is the consent changing depending on the location? But we do not know if

- it is because we got a different version of the APK (that may change from one country to another – different version of the app per country) or if

- it is because of the location of the device when the app is started (this location can be derived from IP address) or if
- its is because of the location of the Google account when the app is started.

6.3.2 Experiment 6: Compare two APKs

With the applications selected from experiment 5 (6.3.1), we compare two installed APKs. In case these files are different, the difference in the consent management flow may be due to the difference in the application source code. On the contrary, if these two files are the same then we will perform experiment 7 (6.3.3) and experiment 8 (6.3.4).

Steps

1. Stop the application on both device A (the United States) and device B (France).
2. We use Android Debug Bridge (adb) to get the installed APKs on device A and device B.
3. Check the MD5 hash of these APKs if they are different.

Possible outcome

For the application, we got different versions of the APK? If yes, it can be the reason for the change depending on the location of the consent flow.

6.3.3 Experiment 7: Change device's IP address and Google account's country

In addition to the results of the experiment 6 (6.3.2), if the APKs of the two versions are not different, we examine whether the difference in the consent settings depends on the device's IP address. In this experiment, we use a fresh account for the Google store to ensure that the account's current country is determined based on the device's IP address.

Steps

1. Stop the app on device A (the United States).
2. Change VPN of device A to France (Google account location will be changed to France and connection IP will be in France).
3. Start the app on device A and check if the consent flow/content changed.

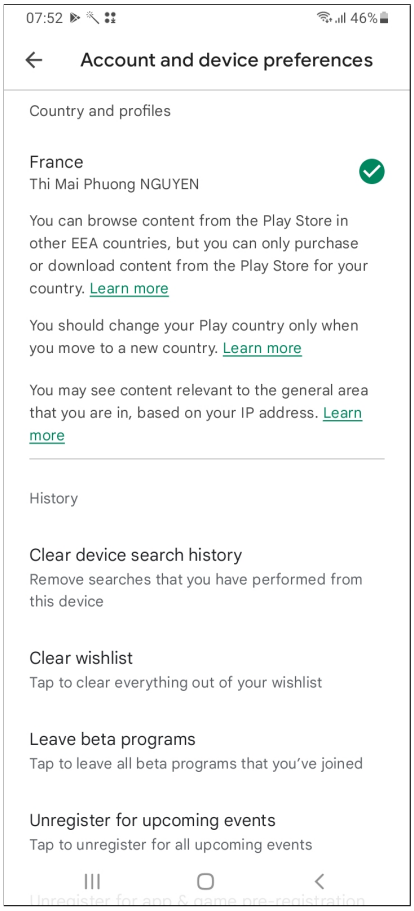
Possible outcome

For the given application, is the consent flow changed? But we do not know if

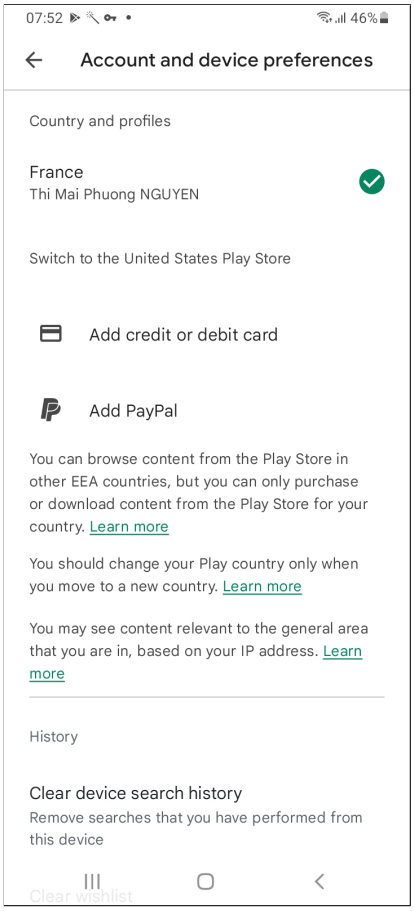
- it is because of the location derived from IP address of the device or if
- its is because of the location of the Google account when the app is started.

6.3.4 Experiment 8: Change only device's IP address

This experiment is similar to experiment 7 (6.3.3) but instead of a very fresh account, we use the Google account which has a payment method in France. According to Google documentation [20], this type of account will not automatically change location, and the only way to change the country for it is to add a payment method for the new country. In case only the device's IP address changes, if the consent management flow of the testing application is still changing, we can assume that the application is using the IP address to decide which laws should be applied to users.



(a) Country of account in normal



(b) Country of account when turn on VPN

Figure 13: Country of Google account which will not automatically update location.

Steps

1. Doing "Factory data reset" on device A.
2. Login to the account ACC_1 which has a payment method in France.
3. Install the testing application.
4. Open the testing application and close it when the consent setting banner show up.
5. Change VPN of device A to United States (connection IP will be in United States).
6. Re-start the app on device A and check if the consent flow/content changed.

Possible outcome

For the tested application, is the consent flow changed? If yes, it is because of the location of the device which is derived from the IP address when the app is started.

6.3.5 Experiment 9: Sign out all Google accounts

This experiment aims to test if the change of the application's consent management flow depends on the field country of the Google account or not. It is an experiment used to consolidate the results of experiment 8 (6.3.4) with the action of signing out all Google accounts on the device.

Steps

1. Stop the app on device A (France).
2. Sign out all accounts on device A.
3. Change VPN of device A to United States.
4. Start the app on device A and check if the consent flow changed.

Possible outcome

For the tested application, is the consent flow changed while the device does not have any Google accounts? If yes, it is because of the location of the device which is derived from the IP address when the app is started.

6.3.6 Experiment 10: Detect the CMP providers domain

As mentioned in the section 5, if an application uses the services of consent management platform providers, the connection to the providers' online servers is necessary. For that reason, we raise the question about the dependence of the change in consent settings of an application on its consent management platform providers' server. It is assumed that the consent management platform provider will have many servers located in many locations. And each server will be in charge of coordinating consent management flow for different laws. In this case, when sending a DNS request, the applications should change the provider server to obtain the necessary consent management flow. Conversely, if the provider server is unique, we can assume that the consent management flow are coordinated only by that server. In other words, the server will rely on the geographic location of the device's IP address to return a compatible consent setting.

With this experiment, we collect the first communications between the application and the providers' server. Subsequently, we used them to consider the role of the consent management platform provider in adjusting the consent settings of the application.

Steps

1. We use the VPN to put device A in the United States.
2. Open the application and capture the network traffics as file NT_1.
3. Stop the application.

4. Turn off VPN on device A, meaning that device back to France.
5. Open the application and capture the network traffics as file NT_2.
6. Compare both files NT_1 and NT_2. Check if the domain name as well as the IP address of consent management platform providers is automatically update follow the location of device's IP address or not.

Possible outcome

Does the application automatically detect the domain name of the consent management platform providers to contact? If yes, does this domain name follow the current device's IP address? If not, are providers the ones who decide which regulations should be applied?

RESULTS OF EXPERIMENTS

In this section, we present the results obtained from performing the experiments proposed in the section 6. In more detail, we completed experiments on 30 separate Android apps as the appendix. They are classified into two sets: Set A contains all applications that can function properly without requiring user login (guest mode); set B contains applications that require user login to function. And for this research, we will use two devices **SAMSUNG Galaxy A32 5G - Android version 12** (device A) and **SAMSUNG Galaxy A71 - Android version 12** (device B) to perform the test.

7.1 Suspicious applications' actions

In the first experiment (6.1.1), we captured the application's network traffics before the user make choices on the consent banner. For each application, we extract a list of domain names from the Pcap file. They are then grouped according to the activity field of the registrant organization. For example, the table 2 is a list of domains that TheFork has connected to before the user makes consent choices.

Table 2: List of contacted domains of application TheFork.

Domain name	Registrant organization	Field of activity
sdk.out.usbla.net	Usabilla	Collect user's feedback
firebaseinstallations.googleapis.com	Google	Tracking/Analytics
firebaseremoteconfig.googleapis.com	Google	Tracking/Analytics
api.thefork.com		
w.usabilla.com	Usabilla	Collect user's feedback
api.lafourchette.com		
api-sdk.datadome.co	DataDome	Detect online fraud
firebase-settings.crashlytics.com	Google	Tracking/Analytics
app-versionning.lafourchette.io		
graph.facebook.com	Facebook	Advertising/Tracking

As can be seen, besides the domains necessary for the application's functionality, many other domains are also connected. And each of those domains belongs to organizations with different fields of activity. In the process of conducting the final results for the experiment, we remove the domains associated with the consent management platform providers as well as the domains directly related to the application's functionality. The result list contains tracker domains of third-party and their appearance frequency in the set of testing applications. The table 3 is the final result we obtained from experimenting on 30 test applications.

From this result, we can see that before obtaining user consent, some applications have initiated connections with many unnecessary servers. For some reason like Certificate Pinning or wrapper tools, we do not know what information was sent to these servers by applications without using a rooted device. However, up to now, many studies focused on analyzing mobile

application violations based on network traffic [36, 26], so we do not go deeper for this experiment. Moreover, the results we got are consistent with the preliminary results in the study by Nguyen et al. [36]. Therefore, we can conclude that the integrated consent management platform apps still have some potential privacy risks for users because of their connections to third-party servers.

Table 3: Top 15 contacted domains by testing apps from the Google Play Store, before any interaction with the consent setting.

Domain name	**Percentage of apps**	**Type of DNS**
firebase-settings.crashlytics.com	70.0 %	Tracking/Analytics
graph.facebook.com	47.0 %	Advertising/Tracking
app.adjust.com	27.0 %	Tracking/Analytics
www.google-analytics.com	13.0%	Analytics
googleads.g.doubleclick.net	13.0%	Advertising/Analytics
ws.batch.com	13.0 %	CRM & push notifications
api-sdk.datadome.co	10.0%	Network/Event Tracking
sdk.out.usbla.net	10.0 %	Collect user's feedback
mobileconfig.sascdn.com	10.0 %	Advertising
asnapieu.com	10.0%	CRM & push notifications
sdk.iad-01.braze.com	7.0%	Collecting data
c.amazon-adsystem.com	7.0 %	Advertising
tags.tiqcdn.com	7.0%	Collecting data
sb.scorecardresearch.com	7.0%	Collecting data
mobile-collector.newrelic.com	7.0%	Monitoring applications

7.2 Consent choices' linked identifier and its storage location

With Set A of test applications that have guest mode, it is obvious that the consent settings will not be attached to the account because they do not require login for guest mode. For that reason, we only perform experiment 2 (6.2.1) and experiment 3 (6.2.2) on Set B of input applications with the purpose to investigate whether privacy settings would be attached to account or device identifiers.

Table 4: The result of experiments which aims to find the linked identifier of consent choices

Application name	**Experiment 1**	**Experiment 2**
Vinted	Required to setup consents again	Consent settings are different
Medscape	Consent setting does not change	Consent settings are different
Veepee	Consent setting does not change	Consent settings are different
Too Good To Go	Consent setting does not change	Consent settings are different
Ryanair	Consent setting does not change	Consent settings are different
McDo+	Consent setting does not change	Consent settings are different

As you can see in the table 4, for most of the apps we are testing, the consent setting does not change when we perform to change accounts. Accordingly, we can affirm that in these applications the consent setting is not attached to the account identifier. Furthermore, we noticed that the consent setting is not synchronized between two devices that use the same account. In other words, with the same account ACC_1 but on different devices, different consent settings exist. Delving into the results of these two experiments, we analyzed the source code of the applications and found that random identifiers were being used to link with

consent choices. Although a lot of consent management platform providers are advertising the consent feature synchronization for users on different devices (cross-device) as well as other platforms (cross-platform) [13, 37] nowadays but with the use of random identifiers, this feature is not yet broadly available in mobile applications.

For experiment 5 (6.3.1), all of the 30 test applications do not keep the prior consent setting as default but will show a fresh consent banner just like the first install. Therefore, we can verify those consent choices are stored in the device's shared preferences as indicated in the section 3.2.

7.3 Global applications and consent setting

As described in the section 6.3, we propose five experiments to find out which fields of information were used by test apps to detect the correct privacy laws that need to comply. In addition, we use the last experiment to discover the role of consent management platform providers in supporting an application to determine the suited laws to apply.

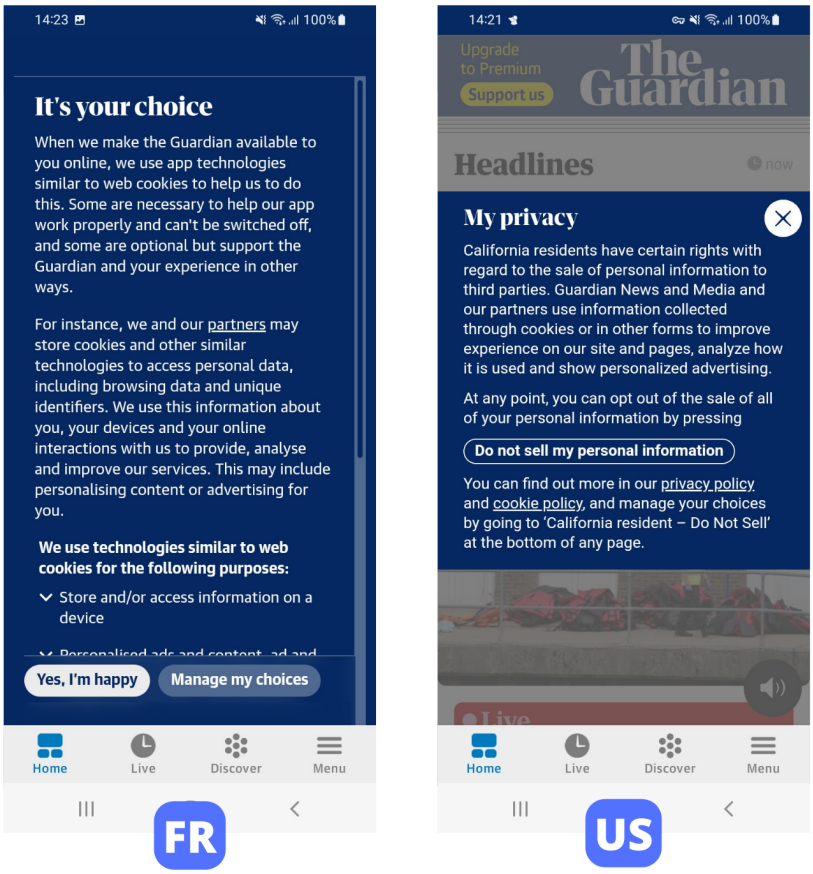


Figure 14: Compare consent setting on two versions of application Guardian

We executed the experiment 5 (6.3.1) on 30 test applications and obtained the following results: 4 apps can't be found on the United State Google store, 21 apps with matching consent management flow on both versions, and 5 apps that meet our selection standards and become test applications for for the next experiments. The list of 5 selected applications is as follows:

1. Booking.com

2. Tripadvisor
3. Flipboard
4. Guardian
5. Vinted

All of these applications have differences in the consent management flow for the United States and French versions. However, what determined this difference still is an unanswered question. To find the correct response, we performed experiments 6 (6.3.2), 7(6.3.3), 8(6.3.4) and 9(6.3.5) with these 5 applications. The final result is presented in the table 5.

Table 5: The result of experiments used to explore the internal process of consent setting

Application name	Experiment 6	Experiment 7	Experiment 8	Experiment 9
Booking.com				
Tripadvisor				
Flipboard				
Guardian				
Vinted				

Consent flow has changed. APK files are different.
Consent flow be the same. APK files are the same.

For all 6 test applications, none of them have 2 different APKs separated for the United States and French Google stores and we can affirm that the adjustment of consent management flow is based on the change of IP address. According to the results of two experiments 7 and 8, the Google account's country field does not affect this modification. Moreover, with the result of experiment 9, we can again confirm that the change of consent management flow is not related to the Google account and the test applications depend on the location of the IP address to update the appropriate law to apply.

Finally, we run experiment 10 (6.3.6) to see whether changing the application's consent settings is reliant on the providers that offer its consent management platform. At first, we filter out all DNS traffic between the applications and the servers of the consent management platform providers. In case the application does not use the services of any provider, we use the main server of the application instead. Then, we compare these DNS traffic between the United States and French versions to reach a conclusion.

Table 6: The comparison of DNS traffic between app and server on two versions US & FR

Application name	DNS query	DNS response (US)	DNS response (FR)
Booking.com	mobile-data.onetrust.io	104.18.32.192	104.18.32.192
Tripadvisor	mobile-data.onetrust.io	172.64.155.64	172.64.155.64
Flipboard	cdn.privacy-mgmt.com	13.249.9.84	13.249.9.84
Guardian	cdn.privacy-mgmt.com	52.222.158.56	13.249.9.4
Vinted	mobile-data.onetrust.io	172.64.155.64	172.64.155.64

When we resembled the results, we found out that the location of the IP address in the DNS response is still stated in a country even after changing the VPN location. For instance, in the Guardian application, the country of the IP address returned in the DNS response at the

connection to the consent management platform provider's server "cdn.privacy-mgmt.com" is always France no matter the location whether the VPN location is changed to the United States. It demonstrates that the contacted server does not depend on the location and the adjustment of consent management flow must come from the IP of the connecting client. We generalize this process as figure 15 below.

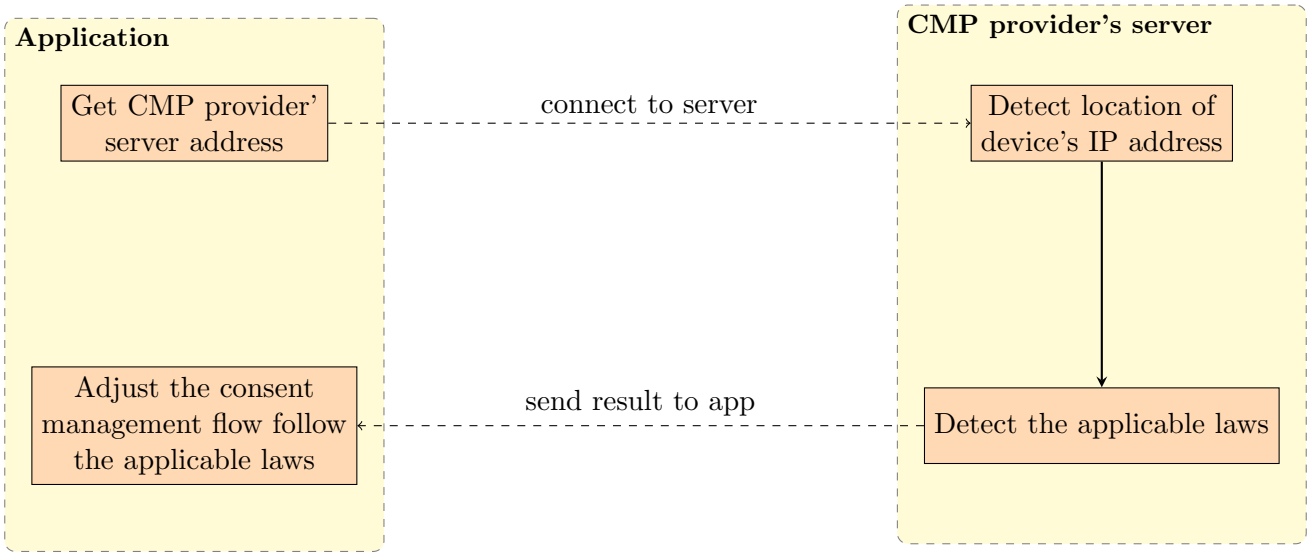


Figure 15: CMP provider and the adjustment of consent management flow

CONCLUSION

In this report, we present the very first overview of the working process of consent management platforms in the mobile ecosystem. With the consent banner, we not only generalize the user interface flow with its involvement but also indicate how user consent choices are stored, processed, and enforced. In addition, we provide a list of approaches that application developers may use to integrate consent management platforms into their products. On the other hand, we came up with indications that can be used to identify consent management platform providers in apps. With applications that have integrated consent management platforms, we finished a few experiments and made our inspection of the potential privacy risks to these apps. Besides, we perform some experiments on applications that have the functionality to change the content and flow of consent management based on the user's geographical location. Hence, we construct a conclusion about the information field used to make those adjustments. And finally, we pointed out the role of the consent management platform providers in determining the applicable law and ensuring that applications will comply with privacy laws.

Future works

The area we want to improve the most in this study is the number of test applications. As you can see, our experiments focused on apps that already integrated consent management platforms. And in fact, there are not many apps on the Google Play store that meet the requirement. For example, in our set of test applications, no candidate has the function of syncing user consent across devices as well as across platforms. Consequently, the information related to this function given in this report is still theoretical but could benefit from a larger scale analysis. These reasons became the driving force for our future work. Automating the process of recognizing applications that already have consent management platforms as well as automating the experiments that we proposed here not only helps to intensify our conclusions but also provides more resources for expanding research on consent management platforms in the mobile ecosystem.

Appendices

A. List of applications

Last updated on 29/08/2022.

Set A - Applications with guest mode

Application name	Package name	Version	CMP Provider
Zalando	de.zalando.mobile	22.10.3	*Self-developing*
Trainline	com.thetrainline	217.0.0.84861	*Self-developing*
Skyscanner	net.skyscanner.android.main	7.71	*Self-developing*
Booking.com	com.booking	33.2	OneTrust
Squid	co.squidapp.squid	2.9.4	OneTrust
Tripadvisor	com.tripadvisor.tripadvisor	48.7	OneTrust
Wallapop	com.wallapop	1.155.0	OneTrust
Doctolib	fr.doctolib.www	3.4.17	Didomi
Apprendre	com.tv5monde.apprendre	3.4	Didomi
AXA Banque	com.axabanque.fr	6.3.2	Didomi
SNCF Connect	com.vsct.vsc.mobile.horaireetresa.android	20220718.2.0	Didomi
Leboncoin	fr.leboncoin	5.83.	Didomi
Idealista	com.idealista.android	9.6.12	Didomi
Flipboard	flipboard.app	4.2.105	Sourcepoint
Le Point	fr.lepoint.android	8.4.16	Sourcepoint
Guardian	com.guardian	6.89.13678	Sourcepoint
Catawiki	com.catawiki2	4.12.0	Usercentrics
weather24	com.wetter.androidclient	2.51.2	Usercentrics
Le HuffPost	fr.huffingtonpost.lehuffpost	1.0.4	Quancast
Cadremploi	com.adenclassifieds.android.cadremploi	5.4.8	SFBX
RTBF	be.rtbf	5.1.5	SFBX
Le Figaro	fr.playsoft.lefigarov3	6.1.18	SFBX
Bonjour RATP	com.fabernovel.ratp	7.14.2	TrustCommander
TheFork	com.lafourchette.lafourchette	20.19.1	Crownpeak

Set B - Applications without guest mode

Application name	Package name	Version	CMP Provider
Vinted	fr.vinted	22.31.2	OneTrust
Medscape	com.medscape.android	10.3.1	OneTrust
Veepee	com.venteprivee	5.30.1	OneTrust
Too Good To Go	com.app.tgtg	22.8.10	*Self-developing*
Ryanair	com.ryanair.cheapflights	3.137.0	*Self-developing*
McDo+	com.md.mcdonalds.gomcdo	5.9.3	Didomi

B. Quantcast functions

```
1  public final void saveAllConsentsPreferences(@NotNull String tcString,
       ↪ @NotNull String nonIabVendorConsents, @NotNull String
       ↪ googleVendorConsents, @NotNull String iabVendorConsents, @NotNull
       ↪ String vendorLegitimateInterests, @NotNull String purposeConsents,
       ↪ @NotNull String purposeLegitimateInterests, @NotNull String
       ↪ specialFeatureOptions, @NotNull String publisherRestrictions, @NotNull
       ↪  String publisherConsents, @NotNull String
       ↪ publisherLegitimateInterests, @NotNull String publisherCustomConsents,
       ↪  @NotNull String publisherCustomLegitimateInterests) {
2      Intrinsics.checkNotNullParameter(tcString, "tcString");
3      Intrinsics.checkNotNullParameter(nonIabVendorConsents, "
       ↪ nonIabVendorConsents");
4      Intrinsics.checkNotNullParameter(googleVendorConsents, "
       ↪ googleVendorConsents");
5      Intrinsics.checkNotNullParameter(iabVendorConsents,"iabVendorConsents");
6      Intrinsics.checkNotNullParameter(vendorLegitimateInterests,"
       ↪ vendorLegitimateInterests");
7      Intrinsics.checkNotNullParameter(purposeConsents, "purposeConsents");
8      Intrinsics.checkNotNullParameter(purposeLegitimateInterests, "
       ↪ purposeLegitimateInterests");
9      Intrinsics.checkNotNullParameter(specialFeatureOptions, "
       ↪ specialFeatureOptions");
10     Intrinsics.checkNotNullParameter(publisherRestrictions, "
       ↪ publisherRestrictions");
11     Intrinsics.checkNotNullParameter(publisherConsents,"publisherConsents");
12     Intrinsics.checkNotNullParameter(publisherLegitimateInterests, "
       ↪ publisherLegitimateInterests");
13     Intrinsics.checkNotNullParameter(publisherCustomConsents, "
       ↪ publisherCustomConsents");
14     Intrinsics.checkNotNullParameter(publisherCustomLegitimateInterests, "
       ↪ publisherCustomLegitimateInterests");
15     setStringPreference(SharedStorageKeys.TC_STRING, tcString);
16     setStringPreference(SharedStorageKeys.NON_IAB_VENDOR_CONSENT_HASH,
       ↪ nonIabVendorConsents);
17     setStringPreference(SharedStorageKeys.OPTION_HASH, Intrinsics.stringPlus
       ↪ (getStringPreference(SharedStorageKeys.NON_IAB_VENDOR_CONSENT_HASH),
       ↪ getStringPreference(SharedStorageKeys.PORTAL_CONFIG_HASH)));
18     setStringPreference(SharedStorageKeys.ADDTL_CONSENT,
       ↪ googleVendorConsents);
19     setStringPreference(SharedStorageKeys.VENDOR_CONSENTS,iabVendorConsents)
       ↪ ;
20     setStringPreference(SharedStorageKeys.VENDOR_LEGITIMATE_INTERESTS,
       ↪ vendorLegitimateInterests);
21     setStringPreference(SharedStorageKeys.PURPOSE_CONSENTS, purposeConsents)
       ↪ ;
22     setStringPreference(SharedStorageKeys.PURPOSE_LEGITIMATE_INTERESTS,
       ↪ purposeLegitimateInterests);
23     setStringPreference(SharedStorageKeys.SPECIAL_FEATURES_OPT_INS,
       ↪ specialFeatureOptions);
24     setStringPreference(SharedStorageKeys.PUBLISHER_RESTRICTIONS,
       ↪ publisherRestrictions);
25     setStringPreference(SharedStorageKeys.PUBLISHER_CONSENT,
       ↪ publisherConsents);
26     setStringPreference(SharedStorageKeys.PUBLISHER_LEGITIMATE_INTERESTS,
       ↪ publisherLegitimateInterests);
27     setStringPreference(SharedStorageKeys.PUBLISHER_CUSTOM_PURPOSES_CONSENTS
       ↪ , publisherCustomConsents);
28     setStringPreference(SharedStorageKeys.
       ↪ PUBLISHER_CUSTOM_PURPOSES_LEGITIMATE_INTERESTS,
       ↪ publisherCustomLegitimateInterests);
29 }
```

Listing 3: Function `saveAllConsentsPreferences` of Quantcast's SDK

```
public final LiveData<String> consentAllAcceptation(@NotNull UIInteractions actionTag, @NotNull Regulation regulation) {
    Intrinsics.checkNotNullParameter(actionTag, "actionTag");
    Intrinsics.checkNotNullParameter(regulation, "regulation");
    String navigationTag = null;
    Boolean acceptAll = null;
    switch (WhenMappings.$EnumSwitchMapping$1[actionTag.ordinal()]) {
      case 1:
        acceptAll = Boolean.valueOf(true);
        navigationTag = String.valueOf(UIInteractions.ACCEPT_ALL);
        tracking.pushEvent(navigationTag, "click");
        return CoroutineLiveDataKt.liveData$default(null, 0L, new UI$consentAllAcceptation$1(acceptAll, regulation, null), 3, null);
      case 2:
        acceptAll = Boolean.valueOf(false);
        navigationTag = String.valueOf(UIInteractions.REJECT_ALL);
        tracking.pushEvent(navigationTag, "click");
        return CoroutineLiveDataKt.liveData$default(null, 0L, new UI$consentAllAcceptation$1(acceptAll, regulation, null), 3, null);
      case 3:
        acceptAll = null;
        navigationTag = String.valueOf(UIInteractions.SAVE_AND_EXIT);
        tracking.pushEvent(navigationTag, "click");
        return CoroutineLiveDataKt.liveData$default(null, 0L, new UI$consentAllAcceptation$1(acceptAll, regulation, null), 3, null);
    }
    acceptAll = Boolean.valueOf(false);
    navigationTag = String.valueOf(UIInteractions.PARTIAL_CONSENT);
    tracking.pushEvent(navigationTag, "click");
    return CoroutineLiveDataKt.liveData$default(null, 0L, new UI$consentAllAcceptation$1(acceptAll, regulation, null), 3, null);
}
```

Listing 4: Function `LiveData` in Quantcast's SDK

References

[1] General Data Protection Regulation (GDPR). *Chapter 3 (Art. 12-23): Rights of the data subject*. URL: https://gdpr.eu/tag/chapter-3/.

[2] Jan Philipp Albrecht. "How the GDPR Will Change the World". In: *European Data Protection Law Review* 2.3 (2016). DOI: 10.21552/EDPL/2016/3/4. URL: https://doi.org/10.21552/EDPL/2016/3/4.

[3] State of California Department of Justice. *California Consumer Privacy Act (CCPA)*. URL: https://oag.ca.gov/privacy/ccpa.

[4] CNIL. *Consent management platform (CMP) ou "plateforme de gestion du consentement"*. URL: https://www.cnil.fr/fr/definition/consent-management-platform-cmp-ou-plateforme-de-gestion-du-consentement.

[5] Rogier Creemers and Graham Webster. *Translation: Personal Information Protection Law of the People's Republic of China – Effective Nov. 1, 2021*. 2021. URL: https://digichina.stanford.edu/work/translation-personal-information-protection-law-of-the-peoples-republic-of-china-effective-nov-1-2021/.

[6] Crownpeak. *Mobile App Implementation Using CaaS*. URL: https://community.crownpeak.com/t5/Headless-Consent-CaaS/Mobile-App-Implementation-Using-CaaS/tap/3253.

[7] Adrian Dabrowski et al. "Measuring Cookies and Web Privacy in a Post-GDPR World". In: *Passive and Active Measurement*. Ed. by David Choffnes and Marinho Barcellos. Cham: Springer International Publishing, 2019, pp. 258–270.

[8] Datareportal. *Digital Around the World*. 2022. URL: https://datareportal.com/global-digital-overview.

[9] Martin Degeling et al. "We Value Your Privacy ... Now Take Some Cookies: Measuring the GDPR's Impact on Web Privacy". In: *CoRR* abs/1808.05096 (2018). arXiv: 1808.05096. URL: http://arxiv.org/abs/1808.05096.

[10] Oxford Advanced American Dictionary. *Third party (noun)*. URL: https://www.oxfordlearnersdic
com/definition/english/third-party.

[11] Didomi. *Consents and Preferences*. 2022. URL: https://developers.didomi.io/api/consents.

[12] Didomi. *Customize the theme & UI*. URL: https://developers.didomi.io/cmp/mobile-sdk/consent-notice/customize-the-theme#custom-notice.

[13] Didomi. *Share consents across devices*. URL: https://developers.didomi.io/cmp/mobile-sdk/share-consents-across-devices.

[14] Didomi. *Third-party SDKs*. 2022. URL: https://developers.didomi.io/cmp/mobile-sdk/third-party-sdks.

[15] IAB Europe. *List of registered TCF CMPs*. 2022. URL: https://iabeurope.eu/cmp-list/.

[16] IAB Europe. *Vendor List TCF v2.0*. 2022. URL: https://iabeurope.eu/vendor-list-tcf-v2-0/.

[17] GDPR. *Lawfulness of processing*. URL: https://gdpr-info.eu/art-6-gdpr/.

[18] Michelle Goddard. "The EU General Data Protection Regulation (GDPR): European Regulation that has a Global Impact". In: *International Journal of Market Research* 59.6 (2017), pp. 703–705. DOI: 10.2501/IJMR-2017-050. eprint: https://doi.org/10.2501/IJMR-2017-050. URL: https://doi.org/10.2501/IJMR-2017-050.

[19] Elizabeth (Liz) Harding et al. "Understanding the scope and impact of the California Consumer Privacy Act of 2018". In: *Journal of Data Protection & Privacy* 2.3 (2019), pp. 234–253.

[20] Google Play Help. *How to change your Google Play country*. 2022. URL: https://support.google.com/googleplay/answer/7431675.

[21] Xuehui Hu and Nishanth R. Sastry. "Characterising Third Party Cookie Usage in the EU after GDPR". In: *Proceedings of the 10th ACM Conference on Web Science* (2019).

[22] Inria. *Inria, an ecosystem*. URL: https://www.inria.fr/en/inria-ecosystem.

[23] Nikhil Jha et al. "The Internet with Privacy Policies: Measuring The Web Upon Consent". In: *ArXiv* abs/2109.00395 (2021).

[24] Ren Jingjing et al. "Bug Fixes, Improvements, ... and Privacy Leaks - A Longitudinal Study of PII Leaks Across Android App Versions". In: Jan. 2018. DOI: 10.14722/ndss.2018.23159.

[25] Thorin Klosowski. *The State of Consumer Data Privacy Laws in the US (And Why It Matters)*. 2021. URL: https://www.nytimes.com/wirecutter/blog/state-of-privacy-laws-in-us/.

[26] Konrad Kollnig et al. "A Fait Accompli? An Empirical Study into the Absence of Consent to Third-Party Tracking in Android Apps". In: *7th Symposium on Usable Privacy and Security (SOUPS 2021)* (June 2021).

[27] Michael Kretschmer, Jan Pennekamp, and Klaus Wehrle. "Cookie Banners and Privacy Policies: Measuring the Impact of the GDPR on the Web". In: *ACM Transactions on the Web (TWEB)* 15 (2021), pp. 1–42.

[28] IAB Tech Lab. *IAB CCPA Compliance Framework*. 2020. URL: https://github.com/InteractiveAdvertisingBureau/USPrivacy.

[29] IAB Tech Lab. *IAB Europe Transparency and Consent Framework*. 2020. URL: https://github.com/InteractiveAdvertisingBureau/GDPR-Transparency-and-Consent-Framework.

[30] CITI Laboratory. *Centre of Innovation in Telecommunications and Integration of service (CITI)*. URL: https://www.citi-lab.fr/.

[31] He Li, Lu Yu, and Wu He. "The Impact of GDPR on Global Technology Development". In: *Journal of Global Information Technology Management* 22.1 (2019), pp. 1–6. DOI: 10.1080/1097198X.2019.1569186. eprint: https://doi.org/10.1080/1097198X.2019.1569186. URL: https://doi.org/10.1080/1097198X.2019.1569186.

[32] Thomas Linden, Hamza Harkous, and Kassem Fawaz. "The Privacy Policy Landscape After the GDPR". In: *CoRR* abs/1809.08396 (2018). arXiv: 1809.08396. URL: http://arxiv.org/abs/1809.08396.

[33] Furini Marco et al. "Privacy Perception when Using Smartphone Applications". In: *Mobile Networks and Applications* 25 (June 2020), pp. 1–7. DOI: 10.1007/s11036-020-01529-z.

[34] Célestin 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: (2020), pp. 791–809. DOI: 10.1109/SP40000.2020.00076.

[35] Lynette I. Millett, Batya Friedman, and Edward Felten. "Cookies and Web Browser Design: Toward Realizing Informed Consent Online". In: CHI '01 (2001), pp. 46–52. DOI: 10.1145/365024.365034. URL: https://doi.org/10.1145/365024.365034.

[36] Trung Tin Nguyen et al. "Share First, Ask Later (or Never?) Studying Violations of GDPR's Explicit Consent in Android Apps". In: *30th USENIX Security Symposium (USENIX Security 21)* (2021).

[37] OneTrust. *Cross Domain and Cross Device Consent*. URL: https://developer.onetrust.com/onetrust/docs/cross-domain-cross-device.

[38] OneTrust. *IAB TCF 2.0+*. 2022. URL: https://developer.onetrust.com/onetrust/docs/iab-tcf-20-android.

[39] OneTrust. *Implementation Approaches*. URL: https://developer.onetrust.com/onetrust/docs/mobile-ctv#implementation-approache.

[40] OneTrust. *Mobile SDK Features*. 2022. URL: https://developer.onetrust.com/onetrust/docs/mobile-ctv#mobile-sdk-features.

[41] Emmanouil I Papadogiannakis et al. "User Tracking in the Post-cookie Era: How Websites Bypass GDPR Consent to Track Users". In: *Proceedings of the Web Conference 2021* (2021).

[42] Quantcast. *Audit Logs*. 2022. URL: https://help.quantcast.com/hc/en-us/articles/360062113654-Audit-Logs.

[43] Quantcast. *IAB Vendor Configuration and Management*. 2022. URL: https://help.quantcast.com/hc/en-us/articles/4411168560407-IAB-Vendor-Configuration-and-Management.

[44] Quantcast. *Mobile Android Implementation Guide*. 2022. URL: https://help.quantcast.com/hc/en-us/articles/1500000450042-Mobile-Android-Implementation-Guide.

[45] Iskander Sánchez-Rola et al. "Can I Opt Out Yet?: GDPR and the Global Illusion of Cookie Control". In: *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security* (2019).

[46] Cristiana Santos et al. "Consent Management Platforms under the GDPR: processors and/or controllers?" In: 12703 (June 2021), pp. 47–69. DOI: 10.1007/978-3-030-76663-4_3. URL: https://hal.inria.fr/hal-03169436.

[47] Zimmec Sebastian et al. "MAPS: Scaling Privacy Compliance Analysis to a Million Apps". In: *Proceedings on Privacy Enhancing Technologies* 2019 (July 2019), pp. 66–86. DOI: 10.2478/popets-2019-0037.

[48] European Data Protection Supervisor. *E-privacy Directive 2009/136/EC*. URL: https://edps.europa.eu/data-protection/data-protection/glossary/e_en#e-privacy_directive2009-136-ec.

[49] PRIVATICS Team. *PRIVATICS - Presentation*. URL: https://team.inria.fr/privatics/.

[50] Usercentrics. *Building your own UI*. URL: https://docs.usercentrics.com/cmp_in_app_sdk/latest/collect_consent/build_own_ui/.

[51] Usercentrics. *The Usercentrics Apps SDK - Module UsercentricsCore*. URL: https://docs.usercentrics.com/cmp_in_app_sdk/latest/#features.

[52] Xiaoyin Wang et al. "GUILeak : Identifying Privacy Practices on GUI-Based Data". In: 2017.

[53] Enck William et al. “TaintDroid: An Information Flow Tracking System for Real-Time Privacy Monitoring on Smartphones”. In: 57.3 (2014). ISSN: 0001-0782. DOI: 10.1145/2494522. URL: https://doi.org/10.1145/2494522.

[54] Razieh Nokhbeh Zaeem and K. Suzanne Barber. “The Effect of the GDPR on Privacy Policies: Recent Progress and Future Promise”. In: 12.1 (2020). ISSN: 2158-656X. DOI: 10.1145/3389685. URL: https://doi.org/10.1145/3389685.

[55] Sebastian Zimmeck et al. “Automated Analysis of Privacy Requirements for Mobile Apps”. English (US). In: *Proceedings 2017 Network and Distributed System Security Symposium*. Proceedings 2017 Network and Distributed System Security Symposium. Korea Society of Internet Information, 2017. ISBN: 1-891562-46-0. DOI: 10.14722/ndss.2017.23034.