CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Subscription Management Platforms under the GDPR

A technical study of Subscription Management Platforms

Master's thesis in Computer science and engineering

Björn Rosengren
Sebastian Sjögren

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2024

Master's thesis 2024

Subscription Management Platforms under the GDPR

A technical study of Subscription Management Platforms

Björn Rosengren
Sebastian Sjögren

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2024

Subscription Management Platforms under the GDPR
A technical study of Subscription Management Platforms
Björn Rosengren
Sebastian Sjögren

Supervisor: Victor Morel, CSE
Examiner: Alejandro Russo, CSE

Master's Thesis 2024
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2024

Subscription Management Platforms under the GDPR
A technical study of Subscription Management Platforms
Björn Rosengren
Sebastian Sjögren
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

In recent times there has been an increase in cookie tracking, where users' data are collected through web cookies. Due to privacy concerns, many regulations have been developed — such as the General Data Protection Regulation (GDPR) —, to regulate information gathering. To ensure compliance with the GDPR, cookies tend to be managed through cookie banners, where users can 1) accept all, 2) reject all, or 3) customize their choice regarding which data can be collected. Recently, there has developed a new cookie paywall, where instead the choices are to either 1) accept all tracking or 2) subscribe to a service to avoid tracking and advertisements. The services providing these cookie paywalls have been named Subscription Management Platforms (SMPs), and the goal of this thesis is to discover what SMPs are technically and legally under the GDPR, and how they relate to standard cookie banners. The results show that SMPs can work as a wrapper to existing cookie banners, where all subscribed users automatically reject all cookies but the non-subscribed must accept all cookies. In this case, the legal responsibility falls to the cookie banner, as the SMP does not handle the consent signal. Additionally, we found that SMPs can collect at least as much information and personal data as regular cookie banners. We also raise several questions about the nature and ethics of SMPs. As SMPs force users who do not pay to accept all tracking, they essentially make privacy a luxury and may increase cookie tracking.

Keywords: Cookies, Cookie tracking, SMP, CMP, GDPR, CNAME cloaking, contentpass

Acknowledgements

We want to say a big thank you to our supervisor Victor Morel, who supported us throughout the work with good ideas when we were lost and important feedback otherwise. We also want to thank our examiner Alejandro Russo for his feedback. Finally we want to thank all of the other researchers in this field, whose work has been vital for this thesis to take place.

Björn Rosengren, Sebastian Sjögren, Gothenburg, 2024-06-26

Contents

List of Figures

1

Introduction

The emergence of the Internet has led to drastic changes in our lives. As infrastructure and technology have evolved, our daily lives have become more and more integrated into the Internet. Hidden from the users is the underlying protocol to enable communication between two parties, the *Hypertext Transfer Protocol* (HTTP). As the use of this protocol grew, so did the usage pattern.

One key aspect was missing for a seamless user experience: the ability to store a state between the parties, which is crucial to allow a server to remember each user specifically. This problem was solved in 1997 when the Internet Engineering Task Force (IETF) introduced HTTP cookies and formalized cookie specifications in RFC 2109 [1]. However, they created something that has been the cause of countless issues ever since, including enormous breaches of security and privacy.

One of the aforementioned issues is cookie tracking, which enables targeted advertising and data mining. Due to privacy concerns about the collection of potentially sensitive information, many regulations have been developed – such as the *General Data Protection Regulation* (GDPR) [2] in the EU –, to regulate what and how information is allowed to be collected by data controllers such as websites.

To ensure compliance with the GDPR, cookies on websites tend to be managed through a standardized cookie banner, where users have the choice of how their data is processed. The choices are "accept", "reject", or user-specified choices of what data is allowed to be tracked and processed.

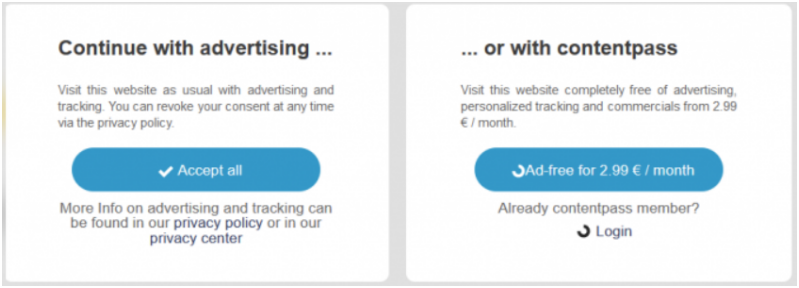

Figure 1.1: An example cookie paywall from the SMP *contentpass*

However, a new form of cookie banner has recently emerged, named *Cookie paywall*, (which can be seen in Figure 1.1). When encountering a cookie paywall, a user is

given the choice to either 1) accept all tracking or 2) subscribe to a service to avoid tracking and advertisements. Therefore a user only has the option to reject tracking if they pay a monthly subscription fee. These services have been named *Subscription Management Platforms*, or SMPs for short. As SMPs have yet to be studied from a technical point of view and are closed source, it raises many questions concerning how they work and whether or not they hold under regulations such as the GDPR.

1.1 Problem

Running a business website relying on advertising and complying with the above-mentioned frameworks can be difficult, as respecting the exact legal clauses is not an easy task. Performing acts illegal under the GDPR can be costly due to fines in the EU: "These types of infringements could result in a fine of up to €20 million, or 4% of the firm's worldwide annual revenue from the preceding financial year, whichever amount is higher." [3]

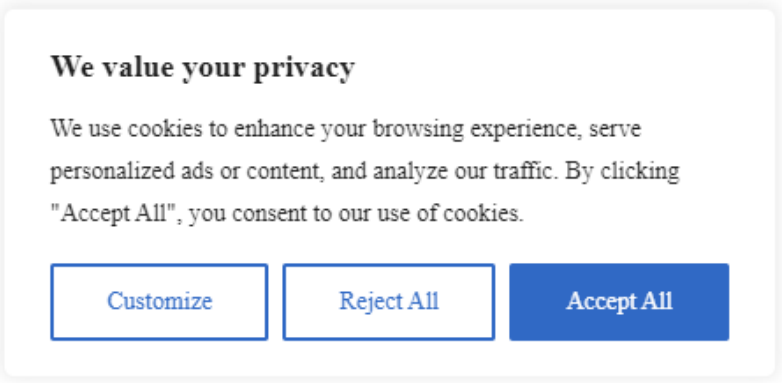

Figure 1.2: A cookie consent banner given by the CMP CookieYes

One solution to help businesses comply with legal requirements is to implement a service that handles users' consent for cookies. These services are referred to as *Consent Management Platforms* or CMPs for short. A CMP solution typically amounts to adding their code to your website which causes a cookie banner to appear. Additionally, they provide a cookie scanner that scans and categorizes the purpose of cookies that advertisers use [4]. In the pop-up banner (which can be seen in Figure 1.2), the user is presented with a list of cookies and their purposes, where the user can accept or deny "non-essential means" [5].

In Article 4 (7) and Article 4 (8) of the GDPR respectively, the actors of *controller* and *processor* are introduced:

> **Controller**: A controller is an entity (person, company, or organization) that determines the purposes, conditions, and means of the processing of personal data. The controller is the one who decides why and how personal data is processed. Controllers have primary responsibility for ensuring that processing complies with the GDPR's principles and requirements.

Processor: A processor is an entity that processes personal data on behalf of the controller. They act on the instructions of the controller and can be any individual, company, or organization that processes personal data on behalf of the controller. Processors are obligated to follow the controller's instructions and take appropriate security measures to protect the personal data they process.

When a CMP takes the role of a *controller*, they must ensure compliance with the GDPR. This involves controlling both the *purposes* and *means* of data processing. In other words, they must clearly specify why the users data is being processed and how it will be processed.

When a CMP takes the role of a *processor*, they must ensure compliance with the controller's instructions of which processing is allowed on the users' data.

Santos et. al. have in 2021 [4] researched the technical (and legal) aspects of CMPs to classify them as controllers and/or processors. SMPs, however, have never been studied from a technical point of view. Therefore, there exists no evidence to classify their role as either *controller* or *processor*, leading to uncertainty in whether or not they comply with the GDPR. However, we know that they are being used, as a recent study found that there exist cookie paywalls on 8.5% of the top 1000 most visited websites in Germany [6]. Specifically, the SMPs *contentpass* and *Freechoice* are mentioned as commonly implemented SMPs. To fill the information gap regarding SMPs, we formulate the following high-level research question:

What are SMPs, technically and legally, and how do they articulate with CMPs?

1.2 Goals

More specifically, our goal is to study SMPs and clarify their functioning to answer the high-level research question. To accomplish this, we start by defining a series of research questions to be studied:

RQ1: What kind of data do SMPs collect, process, or handle? Can SMPs collect data for their own objectives?
RQ2: Do they have access to personal data? How?
RQ3: What is the relationship between SMPs and CMPs?
RQ4: What are the technical evidences in favor of SMPs being controllers and/or processors?
RQ5: What is the relationship between SMPs and publishers?

We want to investigate the technical aspects of SMPs and how they relate to CMPs and publishers. From here we aim for this master thesis to serve as a technical foundation for further legal work surrounding the potential legal and ethical concerns regarding SMPs. This is made easier by attempting to define SMPs as controllers and/or processors, as each has clearly defined obligations and duties in the GDPR.

The core goal is to closely study the technical capabilities of SMPs, the data they collect, and the access they have. We must also compare SMPs with CMPs, to assess their differences and similarities, and attempt to define SMPs under the GDPR. With this information, we should be able to answer our research question.

Due to the many unknown properties of SMPs, a form of exploratory research will be used, where many different ideas and approaches are examined. A significant amount of testing will be performed to see which of the questions can be answered. As many of the questions are connected and overlapped, the findings of some of the questions may help to understand the overall picture. Therefore, one of the core challenges addressed is to design some kind of experiment or method for obtaining information to answer these questions, as there is no known standardized method to use.

1.3 Background

In the following section, we provide the context for where CMPs and SMPs come from and the role they play in the ecosystem of data management on the Web. We also provide some background information regarding the GDPR.

In an effort to protect the privacy of users and communications in the EU, the GDPR [2] and the *ePrivacy Directive* (ePD) [7] were adopted, and enforced in 2016 and 2002 respectively. These legal frameworks strengthen users' privacy by providing the right to privacy of communications and the right to access, rectify, erase, and restrict the processing of their data.

To help the industry to navigate through the legal landscape of these frameworks, the European branch of the *Interactive Advertising Bureau* (IAB Europe) developed the *Transparency and Consent Framework* (TCF) [8]. This framework aims to facilitate compliance with both the ePD and the GDPR when processing personal data and provides a standardized approach for organizations to handle user consent for the use of tracking technology. To ensure legal compliance for websites, some companies have started providing *consent as a service* [9]. This has created a complex environment consisting of many different parties, each serving their own role within data management on the Web. The *actors* known so far in this ecosystem consist of:

IAB Europe An organization that represents the advertising industry in Europe
CMP An organization that provides cookie related services to ensure compliance with GDPR
Advertiser The advertising company that wants information about the users
Publisher A website, potentially hosting a CMP structure
Data subject The user of a website whose data is being tracked

A new consent model has recently emerged where website visitors decide whether to pay a fee or accept tracking, as seen in Figure 1.1. This type of consent model has been given several names, such as *Cookie paywalls*, *consent or pay walls*, *pay-or-tracking walls*, *accept-or-pay cookie banners* etc. This model introduces new

legal challenges in the current ecosystem and introduces a new actor: *Subscription Management Platforms*, or **SMPs**, which adds functionality to CMPs by introducing a payment mechanism in the consent notice.

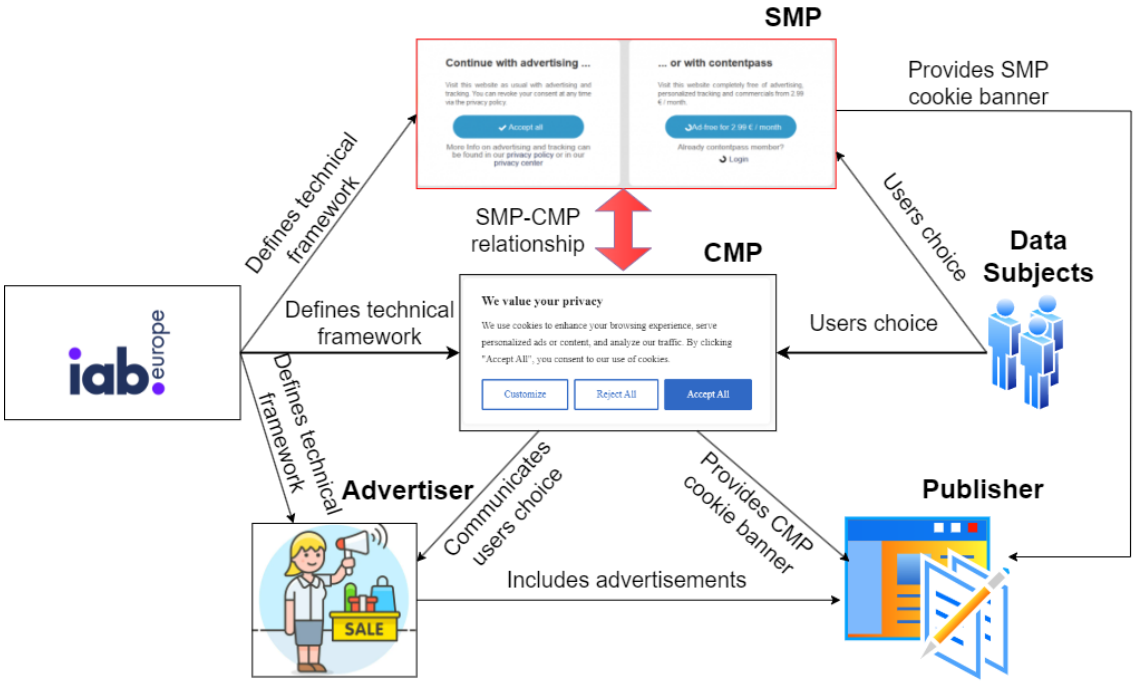

Figure 1.3: The ecosystem of Actors; **IAB Europe** provides a technical framework for advertisers, CMPs, and SMPs, defining how they are allowed to operate. **Advertisers** may include their adverts on publishers and collect data according to users' choices. **CMPs** provide a CMP cookie banner to publishers to allow legal cookie management on the web to collect the users' choices and communicate them to advertisers. **Data subjects** give their choice regarding which cookies can be collected. **SMPs** provides their own cookie banner to publishers, and maintain a currently unknown relationship with CMPs.

The introduction of SMPs raises ample concern, especially considering that CMPs are far from being perfectly implemented. A study by Matte et al. in 2020 [10] found that out of 540 websites using CMPs, 54 % were found to have at least one suspected violation of either the GDPR or the ePD. As CMPs already stand on uncertain legal grounds, Matte et al. discoveries give cause to many new questions surrounding SMPs such as their respect for users' privacy and whether or not we can trust publishers to implement them correctly from a technical standpoint.

The GDPR contains several important articles which relate to CMPs and SMPs. Besides the definition of *controller* and *processor*, it also contains information regarding when consent is considered to be given and what counts as a fair choice. The GDPR defines consent in its Article 4 (11) as:

> "'consent' of the data subject means any freely given, specific, informed, and unambiguous indication of the data subject's wishes by which he or she, by a statement or by a clear affirmative action, signifies agreement

> to the processing of personal data relating to him or her;" [2]

The GDPR defines how a controller is allowed to process data in the recital paragraph 42 as follows:

> "... a declaration of consent pre-formulated by the controller should be provided in an intelligible and easily accessible form, using clear and plain language and it should not contain unfair terms. ... Consent should not be regarded as freely given if the data subject has no genuine or free choice or is unable to refuse or withdraw consent without detriment." [2]

It states that consent is not considered freely given if a user has no genuine or free choice to refuse. This raises the question if the choice an SMP provides to data subjects counts as a genuine or free choice.

2

Related Work

In this section we cover the related work on CMPs and SMPs, but also important work on other related topics and methods relevant to our work. This includes details on cookies, cookie tracking through first-party and third-party tracking, and an increasingly popular method of disguising third-party tracking: CNAME cloaking. Finally, methods of black-box testing are introduced, which allows the testing of software without any knowledge of the system internals, something useful for researching closed sources.

2.1 Prevalence of CMPs & SMPs

Hils et al. [11] examined 4.2 million domains from June 2018 to 2020 to assess the adoption of CMPs over time. They found that the prevalence of CMPs on websites increased twofold in 2019 and once more in 2020, especially on moderately popular websites, due to adherence to EU data protection laws.

Degeling et al. [12] observed the frequency of CMPs on websites in the five months preceding the implementation of the GDPR. They recorded a rise from 50.3% to 69.9% across EU and noted a 16% growth in the use of consent pop-ups after the GDPR was enacted.

Rasaii et al. [6] studied the number of cookie paywalls, used on the Internet by implementing a Web crawler. They found cookie paywalls on 0.6% of 45 thousand queried websites, with a spike in usage found in Germany where 8.5% of the top 1000 websites implement cookie paywalls. Additionally, they found that websites using cookie paywalls send 6.4 times more third-party cookies and 42 times more tracking cookies to visitors compared to ordinary cookie consent banners. They also managed to successfully identify two SMPs that host their cookie paywalls on hundreds of websites, *contentpass* [13] and *Freechoice* [14].

The CMPs and SMPs found are of great interest in this thesis, as they highlight which CMPs and SMPs are the most prevalent on the internet. We have used these articles to find CMPs and SMPs to study.

2.2 CMPs under the GDPR

A study by Nouwens et al. [15] analyzed five widely used CMPs on UK websites to evaluate compliance with laws like the GDPR. The authors revealed that nearly 90% of these consent pop-ups failed to comply with basic legal standards. Additionally, it was found that not including a 'refuse' button on the initial layer of the pop-up increased the rates of positive consent by approximately 22%.

A different study on CMPs has been carried out by Santos et al. [4] which provides a comprehensive analysis of CMPs' functions and obligations under the GDPR, emphasizing the need for clarity in defining their status as data *processors* or *controllers*. Additionally connected to this work are the guidelines that help define the roles of controllers and processors under the GDPR [16]. The authors scrutinize the roles and responsibilities of CMPs in processing personal data, while also shedding light on the use of manipulative design techniques in CMP interfaces and the implications for GDPR compliance.

Santos et al. show how a CMP should typically take the role of a processor, with the *publisher* being the controller. This means the publisher is responsible for determining the purposes and means of processing personal data on their website. They engage the services of a CMP to manage user consent and facilitate data processing activities in compliance with data protection regulations such as the GDPR. The controller provides instructions to the CMP regarding the legal bases, purposes, vendors, and other relevant details for processing personal data on the website.

The authors provide a comprehensive analysis of CMPs' functions and obligations under the GDPR through empirical research and experiments. In their experiments, the authors assume the role of a publisher, where they implemented JavaScript CMP code on an empty website. Observing user interaction with the website, they found that when CMPs act as a processor, CMPs can exceed the boundaries set by the controller. If a CMP extends its operations beyond the scope defined by the controller and engages in additional processing activities without explicit authorization, it may assume a controller role for those specific operations. Therefore, since it determines its own objectives, the CMP qualifies as a controller, which is a breach of its obligations and therefore makes it subject to sanctions (Article 28(10)). [2]

Due to these results, Santos et al. highlight the importance of transparency and legal compliance in CMP operations, calling for enhanced guidelines and oversight by Data Protection Authorities (DPAs). Overall, their work offers valuable insights and recommendations for policymakers and stakeholders to navigate the complexities of CMPs in the realm of data protection and privacy regulations.

This study was selected to be used as a guide to set up a similar experiment for SMPs, as they should be similar in structure. It is useful for connecting any technical evidences we find regarding SMPs with the terms processor or controller which are relevant under the GDPR, and to define any manipulative design techniques we may find among CMP or SMP behavior. Furthermore, many useful terms are established for describing the ecosystem we are working within.

2.3 SMPs

Rasaii et al. [6] discovered interoperability between SMPs and CMPs, finding that the CMP Consentmanager provided integration support for the SMP *contentpass*, which is supported by Consentmanager's documentation [17].

Another related work by Morel et al. [18] has done further research on cookie paywalls and has managed to identify 431 of such paywalls which they analyze and discuss. Among their results, it was discovered that all found cookie paywalls use the "controversial TCF" [18] and that many stood on uncertain legal grounds.

This has been an important motivation for our work, as it highlights that it is very uncertain if cookie paywalls or SMPs are legal. This fact demands more information to be gathered on the topic, and we strive to be among those who attempt to discover the currently unknown attributes of SMPs. The possible integration support could also prove crucial for integrating an SMP and performing experiments using it.

2.4 Deceptive design

Toth et al. [19] discuss the concept of dark patterns, that now go by the name deceptive design, and how CMPs may manipulate publishers into using them. Deceptive designs are design elements or interfaces that are intentionally crafted to nudge users toward a particular choice or action, often by making it more prominent or difficult to opt-out. In CMP interfaces, deceptive designs can be used to steer users towards giving consent without fully understanding the implications, potentially leading to non-compliant or uninformed consent.

By analyzing CMP services on an empty experimental website, the authors identified manipulation of website publishers towards a subscription to the CMPs paid plans and determined that default consent pop-ups often violate the law. Additionally, they showed that certain configurations that are allowed by the CMP may lead to non-compliance with regulations such as the GDPR.

Toth et al. claim their findings demonstrate how CMPs can manipulate publishers into non-compliance, and that it raises concerns regarding the position of privilege CMPs have to influence publishers.

Santos et al. [4] also found several different manipulative design techniques when analyzing CMP interfaces which can influence user decision-making about consent for data processing activities. These techniques can have implications for GDPR compliance, as they may impact the transparency, fairness, and validity of user consent. Some manipulative design techniques observed in CMP interfaces and their implications include:

Misleading Language: Using ambiguous or misleading language in consent prompts can confuse users about the implications of their choices. For example, presenting consent options in a way that downplays the extent of data collection or sharing may

result in users unknowingly agreeing to broader data processing activities.

Hidden Options: Concealing or making opt-out options difficult to find within the interface can hinder users from exercising their right to refuse consent. By obscuring certain choices or making them less visible, CMPs can influence users to accept data processing without fully considering their preferences.

Pre-Selection of Options: Automatically pre-selecting consent options or making certain choices in the default setting can bias users towards providing consent without actively engaging with the decision-making process. This can undermine the principle of freely given, specific, informed, and unambiguous consent required by the GDPR by making consent opt-out instead of opt-in.

Overwhelming Choices: Presenting users with a large number of consent options or vendors in a way that overwhelms or confuses them can hinder their ability to make informed decisions. This can lead to users consenting to data processing activities they do not fully understand or agree to.

All these design techniques fall under the scope of deceptive design, which allows the CMP to manipulate data subjects into consenting through an unclear user interface. As the GDPR has defined requirements of transparent messaging and the fairness of freely given consent, these manipulative design techniques may be illegal under the GDPR.

2.5 HTTP state management

The document *HTTP State Management Mechanism* (RFC 6265) by the IETF [20] defines the syntax and semantics of the HTTP Cookie and Set-Cookie header fields. These headers allow servers to maintain stateful sessions with clients using the stateless HTTP protocol.

The document addresses issues related to privacy considerations, security vulnerabilities, and user controls, and provides recommendations for server and user behavior regarding cookies. It also discusses the historical context of cookies on the Internet and provides guidelines for implementing and using cookies securely and effectively.

2.6 Cookie Tracking

An article by Mayer et al. [21] delves into the landscape of third-party web tracking, shedding light on the privacy implications of third-party tracking.

There are two main types of website resources: A *First-Party resource* originates from the domain the user is currently visiting and any subdomain to this domain. A *Third-Party resource* originates from another domain or IP address.

Similarly, there are *First-Party Cookies* which are cookies that refer to the current host, that is, the website the user is currently browsing. There are also *Third-Party*

Cookies, which are not explicitly set by the current host and are often generated in JavaScript fetch requests or mechanisms similar to another domain. Then if a cookie is attached to the response, the cookie will be set for this particular domain, unbeknownst to the user.

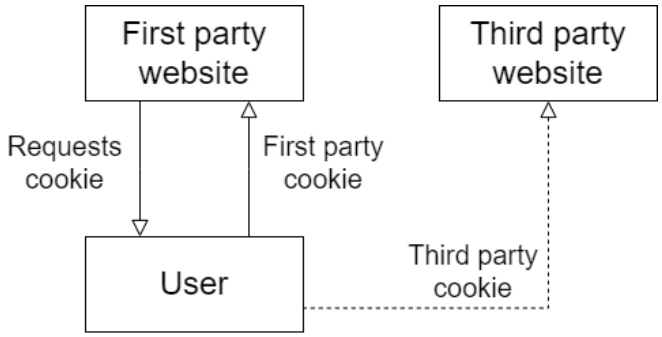

Figure 2.1: The host website requests cookies from a user, who will only send the cookies they consent to. The third-party website however may receive the users' cookies shared with the host website without explicit consent from the user.

The use of *Third-Party Cookies* allows third-party tracking, as a state can be maintained between a user and a domain using a unique identifier. As the user browses between different websites, the identifier is used along with the request the code generated, and thus with information from the origin of the request and the identifier, it is possible to track the user.

Third-party tracking allows entities to collect extensive browsing history information about users, including their location, interests, purchases, employment status, and medical conditions. This data can be used to create detailed profiles of individuals without their explicit consent, raising concerns about data privacy and security.

Furthermore, third parties tracking users' online activities can potentially harm individuals by exposing them to targeted advertising, manipulation, and identity theft. Additionally, users often have limited control over the tracking practices of third-party entities despite survey results showing high levels of opposition to tracking for advertising and analytics purposes.

2.7 CNAME Cloaking

The work of Dao et al. [22] addresses the emerging threat of CNAME (Canonical Name) cloaking, a technique used by tracking providers to evade traditional privacy protections and track data subjects. The authors aim to characterize, detect, and protect against CNAME cloaking-based tracking through a combination of empirical analysis, machine learning techniques, and browser extensions.

DNS records are used to map human-readable domain names like "www.example.com" to various types of information needed to locate and interact with internet services. Examples of DNS records are *Alias* records, which map a domain name to an IPv4 address or IPv6. These records are known as A record for IPv4 and AAAA for

IPv6. There are two apparent problems with these records. Firstly it is because an IP address can change, which then can cause the mapping to point to an invalid address. The other problem is readability. Remembering 4 or 16 numbers is harder to remember than a canonical name, which can result in either bugs or a data leak due to user error. [23]

CNAME records are used to create aliases for domain names by mapping the domain to another domain. This type of record does not suffer from the problems of IP addresses being changed as we are only using domain names. It is possible to chain multiple CNAME records, but the final record has to be an A or AAAA record – a.com -> b.pow -> c.ow -> .. -> A/AAAA record. [23]

CNAME cloaking is a technique used by tracking providers to disguise third-party tracking requests as first-party requests, bypassing traditional privacy protections. The following is a technical description from the work of Dao et al. [22] of how CNAME cloaking works and why it is a privacy concern.

Tracking providers ask their clients to delegate a subdomain for data collection and tracking and link it to an external server using a CNAME DNS record. When a user visits a website, the website embeds a subdomain that resolves to a tracking-related third-party domain using a CNAME record. The browser, unaware of the actual origin of the request, connects to the tracking provider's server and retrieves content along with cookies, which are stored under the domain of the visited website.

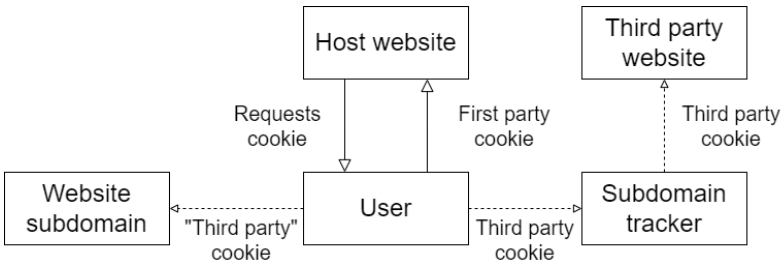

Figure 2.2: A regular website may contain subdomains to handle different tasks. Using CNAME cloaking, a third party may track a user by implementing a tracker as a subdomain on a website. As it is difficult for the user to distinguish the genuine website subdomain from the tracker subdomain this can lead to tracking without the consent or knowledge of the user.

CNAME cloaking therefore makes it appear as if tracking requests originate from the visited website, allowing third-party trackers to collect user data without being blocked by traditional third-party tracking protections.

Dao et al. [22] claim that CNAME Cloaking poses a significant threat to user privacy on the Web by allowing tracking providers to evade traditional privacy protections and track user activities across websites without their explicit consent. Additionally, it is shown that the use of CNAME cloaking has been steadily increasing and a significant number of websites have employed this technique to track user behavior.

In addition, the article discusses how to detect and discover if CNAME cloaking is

being used. One of these methods is the analysis of subdomain prefixes and request patterns, by analyzing the presence of random subdomain prefixes and short requests in the data. Discrepancies in subdomain structures and request patterns between legitimate first-party requests and cloaked tracking requests can indicate the presence of CNAME cloaking. Additionally, the authors developed a browser extension for automatic classification and filtering of CNAME cloaking-based tracking.

We referenced this article to aid in identifying whether or not the CMPs and SMPs we research employ CNAME Cloaking. Furthermore, this article is referenced as evidence supporting the privacy concerns of CNAME Cloaking, motivating research on the topic and a recommendation against it were the technique to be discovered.

2.8 SMPs

Although little is known about SMPs technical details, some research has been performed on their effects and how audiences respond when encountering an SMP. Mueller-Tribbensee et al. [24] discuss reactions to what they call Pay-or-Tracking walls implemented by Meta (Facebook's mother company) and other publishers.

The results show that publishers did not experience a decline in online traffic after implementing Pay-or-Tracking walls. Furthermore, most users opt for the tracking option, while only a few choose the pay option. This means that the choice of installing a Pay-or-Tracking walls results in economic benefits for publishers, as they can track more users and potentially observe a revenue increase of 16.4% [24] compared to using a cookie consent banner (CMP).

The authors find that the price set for the pay option exceeds the advertising revenue that publishers would generate from a user who consents to being tracked. This indicates that users seeking privacy have to pay a premium compared to the revenue generated from tracking them.

The authors also voice some ethical concerns with Pay-or-Tracking walls, for instance, that privacy may become a luxury only for those who can afford to pay for it. This could create a digital divide where only certain users have the financial means to protect their privacy. Additionally, privacy activists question whether the payment required by pay-or-tracking walls aligns with the concept of freely given consent [25]. Users may feel pressured to pay for privacy, raising questions about the voluntariness of their choice.

In general, the study suggests that despite the fact that Pay-or-Tracking walls raise privacy concerns, this does not lead to a decline in online traffic. Therefore, SMPs can be a profitable strategy for publishers, with most users opting for tracking and overpaying for privacy.

2.9 Black-box testing

Nidhra et al. [26] define black-box testing as when a tester has no internal knowledge of an application, making the application a black box from the perspective of the tester. They also outline several different methods of effective black-box testing.

One possible method of black-box testing is to manually browse websites containing a certain application. By experimenting with the websites and by viewing any interactions that occur, a tester may learn about the application. The strength of this method is its simplicity, while its weakness is that it relies heavily on the experience of the tester. To achieve optimal results, the exact methods of the manual testing have to be fitted to the task and are therefore application dependent.

When testing CMPs and SMPs, much relies on being granted access to these services. However, in the circumstance that no access is granted, black-box testing could still be performed to gather as much information as possible without having access. Studying the methods of black-box testing could therefore be crucial to allow testing for the CMPs and SMPs we do not gain access to in our thesis work.

3

Methods

The following chapter describes our method of research. This includes any software and hardware requirements, as well as our approach to gathering data regarding SMPs and CMPs.

To understand the differences and nuances between CMPs and SMPs, we first experiment with CMPs. We then experiment with SMPs to understand how they relate and what their capabilities are from the perspective of publishers and users.

For testing CMPs, we took great inspiration from the paper *Consent management platforms under the gdpr: Processors and/or controllers?* written by Santos et al. [4]. The details of this paper concerning the technical structure of CMPs helped to design the software and the analysis of CMPs. To help with cookie management, we used the IETF guidelines [20] on *HTTP State Management* as a guideline when choosing applications and writing our code.

As cookies are application-dependent, i.e. the processing and handling of cookies can vary, some considerations are needed to consider what applications we use. A high-level requirement is to simulate a realistic real-world scenario of a user browsing a website, therefore, it needs to be able to execute JavaScript or dynamic events after the website is loaded.

3.1 Software

The programming language *Python* is our main tool to create our software stack. We chose this language due to its rich ecosystem of modules and as such, reduce the boilerplate code to a minimum.

As we expect to use different operating systems (OS) during development and deployment, we use *Docker* to avoid creating specific code suitable for each OS. This allows for an easy and portable setup and allows us to abstract the hardware away as long as it can run Docker.

3.2 Software stack

We wanted a realistic scenario of a simple infrastructure of a small company. We chose to use *NGINX* as the bordering software where initial requests are sent which are then routed by predefined paths depending on the pattern match of the requested host. This hides the underlying software behind NGINX which disguises our intentions.

The IETF guidelines specify a cookie attribute called the **Secure** flag, which only allows handling cookies if the channel is secure. We use Transport Layer Security (TLS) to secure and encrypt the channel between the server and the user. Certificates for *TLS* are generated from the non-profit foundation *Let's Encrypt* [27] and are then added to NGINX.

Hosting the website is based on a *Flask*-project, a web framework using Python. As we work remotely with our servers and the connection is secured under TLS, all tools that listen to networking traffic between the server and a user are ineffective due to encryption. To gain further insight into the requests executed from the browser, we created a *sniffer*-module using real-time communication protocol *WebSockets* to send and receive data. This module implements an observer pattern to make the code independent of the number of listeners of the module. This module is attached as an intermediate layer to our web server. We use this module to extract the entire HTTP request. Listening is done by connecting to the sniffer using our client and the captured requests are sent to all connected listeners. As the extracted data might be sensitive, we made an additional route in NGINX for the WebSockets and reused TLS. Figure 3.1 shows the information flow within this stack. We use this software stack on our servers, where it is configured to suit our needs for a particular task.

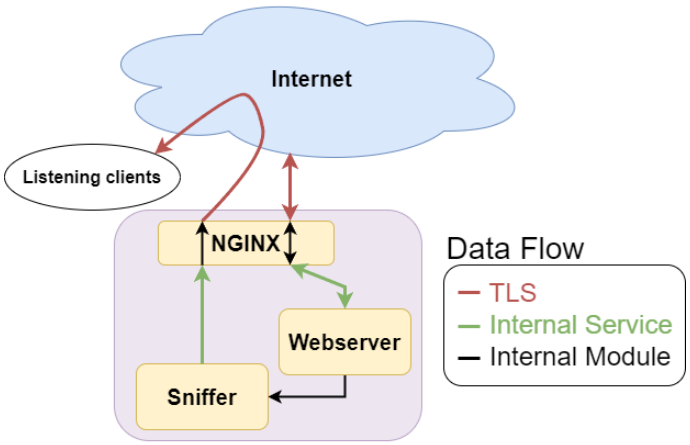

Figure 3.1: An overview of the software stack. A user connects to the sniffer using our client to listen to all requests on the server. The yellow boxes are the services hosted on the device (purple box). The red arrow describes the TLS-secured channel. The black arrow describes the internal channel within a service. The green arrow describes channels between services.

3.3 CMP

Testing the CMPs, we first test *InMobi Choice* (formerly known as *Quantcast Choice*) as this service was one of the most popular CMPs according to [11], and their service is free to use without requiring a meeting to gain access. We create an account and use their supplied code which we add to our website.

In the *InMobi Choice* admin page, we are given some settings and tools. We are allowed to select different banners to comply with multiple regulations and frameworks. The platform also lets us choose which vendors should be allowed or disallowed. We opted for default settings and a cookie banner that complied with TCF 2.2.

As we wanted to gain a better understanding of CMPs and how they operate from a non-user perspective, we wanted to test additional CMPs. This gives us better insight into their techniques of loading their banner and understanding the general differences between the different CMP actors. However, all other services listed in [11] required a meeting or similar, or payment to gain access. We then searched for CMPs by reviewing the related works in section 2.1 and additional searching with Google. We select *CookieYes* because they allow us to register to gain access. As with InMobi Choice, we use the default settings on CookieYes.

3.4 SMP

SMPs are comparatively more difficult to study than CMPs mainly due to one reason: accessibility. SMPs operate under a more controlled partner deal, where only partnered publishers are allowed to use their service on their websites.

Ideally, we would successfully partner with each of the SMPs we desire to study, in reality, this may not be the case. We contacted **contentpass** and **Freechoice** to request access to their service for research and testing. Only contentpass replied and after a meeting with their CEO, we gained access to their service.

We follow contentpass guidelines to implement their SMP on our website for further testing and evaluation. Comparing the setup process of the CMPs, contentpass requires us to configure our DNS by linking their services with a CNAME as a subdomain and selecting a CMP from a given list of partnered CMPs. We then run the same experiment as the CMPs and conduct additional testing with third-party resources to observe the results.

The list of CMPs from contentpass, all of which required payment for access. After contacting all CMPs on the list, CCM19 and Consentmanager responded and gave us access to their services for a limited time. CCM19 and Consentmanager provided a similar admin interface to the two other CMPs mentioned.

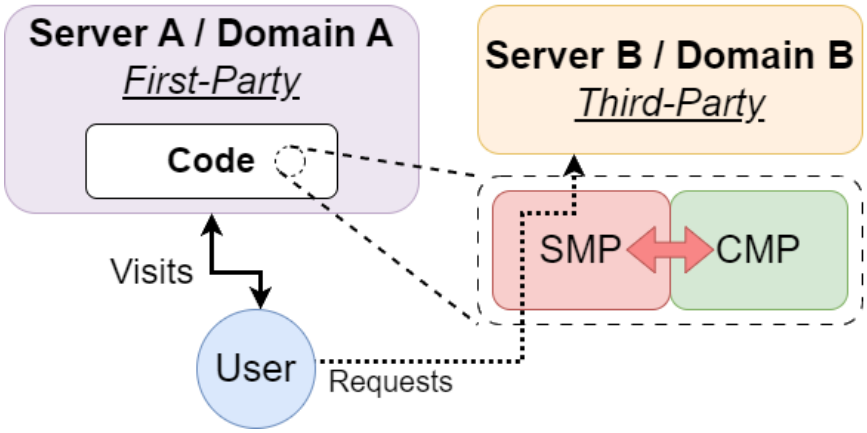

Figure 3.2: An overview of the information flow between the different entities. After a user receives the response from **A**, the SMP/CMP should handle the consent given by the user. If the user selects to allow being tracked, then requests will be sent to third-parties. The red link between SMP and CMP is the relationship.

3.5 Data gathering

As the cookies should be isolated between domains, we use two domains to test first-party and third-party cookies. Initially, we only had access to subdomains of Domain A (DA) and Domain B (DB). Later, we gain access to a domain that we use solely for server A to test whether subdomains can affect the results.

We configure the DNS by pointing server A to the DNS records of DA, while server B is configured to point to the DNS record of DB, and a subdomain of DA to test CNAME cloaking. This allows us to see whether the CMPs and/or SMPs have the technology to detect this technique of masquerading as a first-party resource. An overview of the information flow of data between the different actors can be seen in Figure 3.2.

Loading a website usually happens in two steps. The first step is when a user contacts the server and receives the code to load the website. The second step consists of when the browser executes the code. The HTML-tags that contain an attribute *src* – *img*, *iframe*, and *embed* – which point to a domain that the browser reads to fetch the resource during execution. This occurs without any input from the user. Furthermore, if the user has JavaScript enabled, then HTML-tag *script* will be loaded and potentially executed.

When the execution of the received code occurs at the user, unless there is an HTML-tag that points back to the visited server, this server should not receive any additional requests. Therefore we need to test the flow of data from the perspective of a user. The method of testing user interaction on the website is conducted on web browsers – Firefox version 124.* and Chrome 123.* – using their supplied developer tools (DevTools) open to log all network activity. Both browsers were run in incognito mode with third-party cookies enabled, using default settings, and without additional extensions.

We created a blank website with a CMP or SMP/CMP implemented. Understanding the network events that occur during the load of a website can give hints of the implementation. Through practical testing on our server and gaining a deeper understanding, we learned that the different implementations have unique signatures in network traffic and the shape of the data fetched/sent. We can use these signatures to see how it is implemented and identify what SMP or SMP/CMP combination is used. For this reason, we chose to conduct black-box testing on other websites to get more information on how SMPs work. Performing black-box testing, we use our browsers *DevTools* to inspect the websites, analyze the network traffic, and examine the data sent and received.

Based on the findings from [18], where websites were crawled to identify the prevalence of cookie paywalls, a list of sites with detected cookie paywalls was compiled. From this list, we perform manual black-box testing on six websites that do not contain either *contentpass* or *Freechoice*. Specifically, we tested 2 instances of each website for *contentpass* and *Freechoice*. We choose not to extend our research beyond these websites as the results of this black-box testing prove sufficient for this thesis.

3.6 Third-Party resources

The CMPs and SMPs often deploy their crawlers to scan your website and ensure compliance. These crawlers scan for loaded resources and notify the website owner of potential compliance breaches. We only tested Consentmanager and *contentpass* as we could not get the other crawlers to work.

Investigating the effectiveness of these tools, we aim to test how these would react when using third-party resources on our website. To achieve this, we set up an additional server **B** with a similar configuration as A but instead to respond with a resource – such as cookies, JSON object, or plain text – using both JavaScript and the mentioned HTML-tags in the previous section.

Additionally, we also fingerprint – unique patterns an object exhibits – our browsers using our JavaScript code to see how the scanners would react when extracting information in an HTTP-GET request using path or query arguments. We use web browser documentation on their API to find what information we can gather, then do a fetch request containing a JSON object.

4

Results

In this chapter, the results of our experiments and research are presented for both CMPs and SMPs. For each result, we will refer back to our research questions and specify which research questions it helps to answer.

4.1 CMPs

We present in this section the results from the analysis of the CMPs InMobi and CookieYes, along with CCM19 and Consentmanager that we used with *contentpass*.

We researched CMPs to better our understanding of how they work in order to understand SMPs. We also searched for information that may help in answering RQ1 "**What kind of data do SMPs collect, process, or handle? Can SMPs collect data for their own objectives?**" and RQ2 "**Do they have access to personal data? How?**" Depending on the relationship between SMPs and CMPs it may be possible for an SMP to access this information, meaning this would then be informative of how and what is accessed. Additionally, understanding CMPs is important for RQ3: "**3 What is the relationship between SMPs and CMPs?**", as the relationship cannot be fully understood if both parties have not been researched.

4.1.1 InMobi

The InMobi documentation shows what vendors are allowed or disallowed on their admin panel, and by default, all the vendors are allowed. They also offer a crawler to scan your website, but this is only available for mobile apps – Android, and iOS.

We found that when the *InMobi Choice* cookie banner loads, it generates session data and then fetches their endpoint using query argument with the session data in serialized JSON format, which can be seen in Listing 4.1. The information contains a timestamp when an event occurred, a unique session ID, and other metadata such as versioning and what directive the banner should follow.

```
{
  "accountId":"{Redacted account ID}",
  "domain":"{Redacted domain A}",
  "publisher":"{Redacted domain A}",
  "cmpId":10,
  "cmpVersion":"2.52",
  "displayType":"tcfui:mandatory",
  "configurationHashCode":"XWC/ICbo7Y3eou2cHQTvNg",
  "tagVersion":"V3",
  "gvlVersion":2,
  "clientTimestamp":1710765402741,
  "operationType":"init",
  "sessionId":"GDPR-auuje1zdgpq2u1l3rhe1"
}
```


Listing 4.1: The Initial request from InMobi Choice, where the information relevant to a data subject is the timestamp of the entry and the unique ID (sessionId) of the data subject.

When a user interacts with the cookie banner, a fetch request is generated, containing information on the event in the Listing 4.2 below. This information contains a timestamp when the event occurred, where the user originated, the destination, and the generated session ID from the initial request. This means the CMP can send some information about a data subject before consent is given.


```
{
  "userEvents":
    [
      {
        "clientTimestamp":1710765402740,
        "event":"startOnPage:GDPR_0"
      },
      {
        "clientTimestamp":1710765439736,
        "event":"goToPage:1"
      }
    ],
  "clientTimestamp":1710765439736,
  "operationType":"navigation",
  "sessionId":"GDPR-auuje1zdgpq2u1l3rhe1"
}
```


Listing 4.2: Upon interacting with InMobis cookie banner the following data is sent after clicking on *Options*. The data contains a timestamp of the event, which button was pressed, and the unique ID (sessionId) of the data subject clicking the button.

This information may help in answering research questions 1. and 2. as it is

informative regarding what is accessed and how.

4.1.2 CookieYes

Compared to InMobi, the documentation of CookieYes only states that they follow the IAB TCF framework. Using their banner, the only noteworthy result we found is that they show on their admin panel the unique ID generated for a user, similar to how InMobi also uses unique IDs. They offer a crawler to scan the website for any cookies. The result is for the publisher to classify the cookies manually.

4.1.3 CCM19

CCM19 is one of the CMPs currently partnered with *contentpass* to integrate with their service. However, payment was required for access to the CMP to allow integration with *contentpass*. Despite the free trial given by CCM19, this service plan did not contain a crucial setting regarding "ensuring your website is ad and tracking-free", which was necessary for a proper configuration enabling integration with *contentpass*.

CCM19 was therefore only usable once a proper version had been supplied by the company behind it, and it was then used with additional code given by *contentpass* which takes precedence and control over the code supplied by the CMP. However, in their code, they contact a CNAME cloaked domain belonging to *contentpass* to fetch *contentpass* driver code. consequently, this enables the extraction of cookies if the cookie-attribute *domain* allows for subdomains; SMP can gather more data than they should.

As with InMobi Choice, hundreds of vendors are given by default when under the IAB TCF framework. Using CCM19, the supplied admin panel they have a page similar to CookieYes where CCM19 shows logs of unique IDs of visitors that have selected *accept all*.


```
{
    "ucid":"57e6fa1aa59b118e",
    "lang":"en_US",
    "clientUserAgent":
        "
            Mozilla/5.0 (Windows NT 10.0; Win64; x64)
            AppleWebKit/537.36 (KHTML, like Gecko)
            Chrome/125.0.0.0
            Safari/537.36
        ",
    "clientOs":"Win32",
    "clientLang":"sv-SE",
    "actualUrl":{Redacted domain A},
    "actualRef":"",
    "actualOpened":1
}
```

Listing 4.3: Upon interacting with *contentpass* consent banner when using CCM19 as CMP. The following data contains fingerprinted information about a user such as what OS they are using and the browser. This is generated and sent after clicking on *Privacy settings*.

Comparing this CMP to the others, this CMP offers a more extensive control of the resources on the website. Both InMobi and CookieYes offer only cookie-related tools, CCM19 adds to this by enabling the blocking of resources, A/B testing, and a crawler to scan for any resource on the website.

Their method of blocking resources is for the publisher to change the name of the HTML-tags that use *src* attribute, this stops the browser from fetching the resource when the website loads. Stopping the browser from executing script-tags containing JavaScript code, the HTML-tag *type* is changed to a predefined value.

The A/B testing is to test two different consent banners between two randomly assigned audiences on the same website. This testing gives two different consent banners, which makes it harder to identify the CMP used by looking at the consent banner.

We could not make the crawler detect our code fetching data from a third-party website or detect our script that sets a cookie. This could be a user error and needs further testing.

During testing, we found that as with InMobi, interacting with the consent banner causes it to generate an event containing some data. The Listing 4.3 shows the data generated in the event. This data contains a unique ID, and fingerprinted information about the user such as which OS they are using, what address they are browsing, and what browser they are using.

4.1.4 Consentmanager

This CMP provides similar tools to CCM19 but offers more advanced A/B testing capabilities. Instead of being limited to two designs of the consent banner to test, they allow for any number of designs. These designs are then randomly assigned to visitors.

When loading the consent banner, similar to CCM19, requests are sent out to *contentpass* CNAME cloaked domain.

Blocking of content is done similarly to CCM19 by renaming the mentioned attributes to prevent the loading of HTML-tags. The major differences are the predefined names of the attributes in the HTML-tag, and additional attributes specific for their implementation. Some attributes give the publisher more fine-grained control of what should be blocked.

Their crawler detected the different HTML-tags that fetched a resource and alerted

us. We had to take action and classify or block these resources. This applied to first-party, CNAME cloaked, and third-party resources.

4.2 SMPs

In our thesis, *contentpass* was directly studied as we gained access to their service. We also encountered unknown SMPs for which we could not determine the origin from either DNS lookups or from the actual domain name. For these unknown SMPs and *Freechoice* we chose to blackbox tested.

4.2.1 contentpass

At our request *contentpass* provided access to their SMP, creating an environment where we became de facto partners of *contentpass*. This access allowed us to study their code and implement the SMP on our test website.

To use their service, they required us to configure a DNS CNAME record as a subdomain to our domain to fetch their JavaScript code containing their consent banner from their servers. By utilizing a subdomain, it looks like it originates from our website and they have access to set cookies for the domain and subdomain as it is effectively a first-party cookie.

They also tell you to block any third-party resource on your website by implementing the techniques your selected CMP uses to block content. To verify that no third-party resources exist on your website, they utilize a crawler to detect these resources.

We implement the partnered CMP to our website with the supplied code. We only modify the given CMP code when instructed to do so. From testing their implementation, it seems to operate by using the CMP consent banner. Their instructions to implement the SMP are to disable all other functionality of the CMP consent banner such that a user is limited to either select “allow all” or “subscribe”. From a technical point of view, the consent banner can now either send a consent signal and activate all disabled resources, or disable itself if the user chooses to subscribe.

Following these steps, the blocking worked as expected. When a user accepts to being tracked, your browser starts loading resources from third-party resources, and subscribing results in some requests to the subdomain *contentpass* maps to.

Adding our JavaScript code using the standard fetch-function to send a GET-request to server B without CNAME cloaking, resulted in their scanner giving us alerts that there were unknown third-party resources on our website. To suppress these alerts, we can allow these resources by adding them to an allowlist. Adding server B to the allowlist, a modal states the responsibility of the origin of this third-party resource. This can be seen in Figure 4.1.

We tested to utilize CNAME cloaking. We found that their scanner could not detect this technique. We managed to set any cookie, load any resource, and send any data

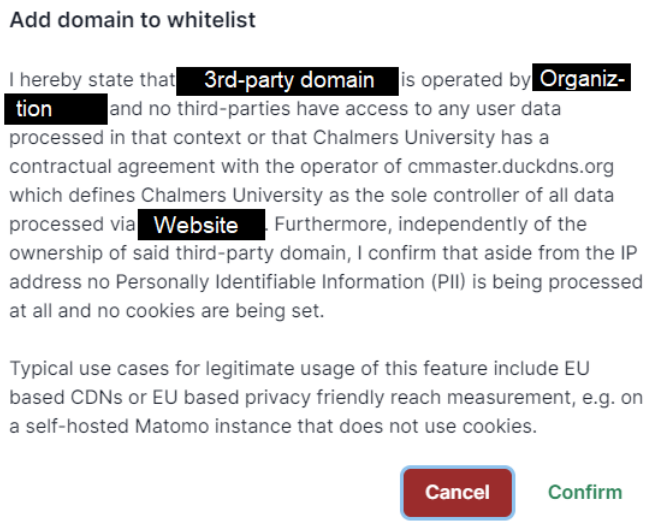

Figure 4.1: Modal given when trying to load a third-party resource on our website

without their scanner alerting us of any non-compliant resource.

When using *contentpass* with CCM19 as CMP, some additional events containing fingerprinted data from a user's browser are generated when clicking on *Privacy settings* on the consent banner. The destination of these requests is the CNAME subdomain that *contentpass* requires you to configure. These events sent a JSON object, which can be seen in Listing 4.4. The notable data this object contains are: unique ID (*cpabid*), timestamp when the event occurred (*cpts*), what part of the banner the user interacted with (*ea*), and fingerprinted browser data of the size of the viewport (*vp*).

Using *contentpass* with Consentmanager as CMP did not exhibit these events with fingerprinted data when interacting with the consent banner.

```
{
    "cpabid":"3beb66ed-9038-4e57-a758-8697674fb289",
    "cppid":{Redacted account-id},
    "cpamp":false,
    "cpbf":1,
    "cpbotr":0,
    "ec":"wall",
    "ea":"cmp",
    "cpts":1716339438777,
    "cpdf":true,
    "cpvt":3851,
    "cpdr":"other",
    "vp":"609x919"
}
```

Listing 4.4: Interacting with *contentpass* consent banner, the following JSON data is sent after clicking on *Privacy settings*.

4.2.2 Blackbox testing SMPs

From a random selection of websites, manually browsing websites yielded mixed results and showed that identifying which SMP a website uses can be difficult. From interacting with CMP services, we noticed that some let you pay for a higher tier of service. The higher tier allows the modification of the consent banner to delete any eventual mentions of the company that provides the solution, making it hard to know if the consent banner was originally from a CMP or a different source. Furthermore, many consent banners have the same layout and shape, increasing the difficulty in determining the SMP provider.

The data gathered from the black-box testing is where the different requests originate, the shape of the received or sent data – cookies, JSON objects, HTML-code, etc. –, and the JavaScript code for CMP/SMP. Additionally, we found that the different SMPs and CMPs have different naming schemes and that they differ in how they structure the stateful data on a user's browser.

Further findings from the network data showed that some SMPs utilize CNAME cloaking. The name of the subdomain that uses CNAME varies from site to site, but some sites use the same name. We also saw that not all SMPs use CNAME cloaking, with Freechoice being an example. The services that exhibit the usage of CNAME cloaking had a different structure – keys and values – of the sent data, hinting that the underlying mechanism might use CMP not unlike *contentpass* to handle the consent banner.

Many consent banners also log events when a user clicks on a link that "updates" the banner, for example, when you want to read what you consent to. The data from these events has a similar structure to *contentpass* JSON data with unique IDs and from which page a user originated.

4.3 Additional Findings

As part of our research, we also found several things not directly tied to our research questions that we found interesting and worth sharing.

4.3.1 Subdomains

During empirical testing, we noticed that subdomains do have some limitations. Some services do not accept subdomains as your primary domain, which can prohibit what can be tested. When used with CMPs and/or SMPs, there were no issues following their guide to install their service to your application, and we did not observe any difference in behavior between hosting an application using a subdomain or a domain.

4.3.2 Web browsers

During testing of the crawlers, we wanted to gather as many fingerprints regarding a user's browser as possible and send the data to server B.

Without CNAME cloaking and with blocking enabled, we tried to use query arguments to extract the data from the user. Both Firefox and Chrome blocked these requests due to "mixed content blocking". However, we were able to extract the same data on both browsers by utilizing the url-path as an escape hatch. We did not manage to set any cookies as these are classified as third-party.

In contrast, by utilizing CNAME cloaking, the browsers did not block query arguments or url-path. This allows both methods to extract data and subsequently set any cookie for the domain and subdomains.

4.3.3 Deceptive design

Upon installing a CMP on our test website, we were given a selection of options to choose from when designing it. One possible design is to make rejecting cookies harder, by hiding it under an options menu. You could also make the cookie banner opt-out instead of opt-in by forcing "accept all" as already selected. This would make the cookie banner contain deceptive designs, a term describing different design techniques to manipulate users, in this case towards giving consent. However, a different design is possible, where a user can easily reject all or accept all cookies. This type of design contains no defined dark pattern. This means that some CMPs provide the publisher an opportunity to implement deceptive designs in the cookie banner, but does not demand it.

Many of the CMPs we studied use various implementations of deceptive design. However, if these deceptive designs are used seemed to somewhat depend on the publishers when installing a CMP. When researching SMPs, we could not find any results defined as deceptive design.

5

Discussion

This chapter contains a discussion of the results found and how they relate to our research questions. We will analyze the results, compare them to each other, and discuss the ethics and legality of CMPs and SMPs in relation to the GDPR. We also mention the limitations of this thesis.

5.1 SMP

This section discusses the capabilities of SMPs and how they relate to CMPs.

5.1.1 The model

Testing *contentpass* and using their service and tools gave us great insight into their model. They do not implement their own CMP but rather refer to a list of CMPs that they have partnered with. Installing the SMP code required us to select one of the CMPs and then follow a setup guide to implement their given solution. Using their service and tools gave us great insight into their model, and how it relates to other SMPs.

Using black-box testing, we could observe the network traffic from the randomly visited websites, and compare it to *contentpass* to see how they differ. There were some websites where we could not identify the SMP, but through testing, we could roughly identify how their implementation works. This evidence points towards the use of a model where the SMP extends its functionality around an existing CMP to enable payment mechanisms. When a data subject is shown the modified CMP consent banner, they are looking at either "accept all" which signals the underlying CMP to take control and activate fetching of resources, or pay which disables the shown consent banner. However, this is not always the case as we found that not all followed this model. Another SMP, *Freechoice*, seems to implement a more integrated solution where instead of partnering up with CMPs, they provide a CMP with a payment mechanism.

A design principle in programming is code reuse. Reusing existing solutions saves a lot of time and effort and reduces the complexity of the entire module as the different components can be tested more individually. Using existing solutions also more likely ensures that the solution works. Since some SMPs use existing CMP solutions, the

responsibility moves towards the CMP when handling the consent signal and the blocking of resources.

The other SMPs we found also utilized CNAME cloaking to disguise itself as a first-party resource. By implementing itself through CNAME cloaking, it looks like a subdomain on a publisher's website. This can be seen as a manipulation tactic, similar to deceptive designs, to convince data subjects of their trustworthiness to avoid being blocked by various adblockers.

5.1.2 The subscription model

At first glance it seems we have a choice: either accept to be tracked or pay a monthly fee to avoid all tracking. This subscription model has a problem: when someone subscribes to an SMP, they are forced to give some additional information to do so. For a *contentpass* subscription, the information required is an email address, password, country of residence, and payment method. In the case of a billing address, a name is also required [28]. This means that when visiting a website implementing an SMP, browsing the website anonymously is impossible due to cookie paywalls.

If an SMP subscription is enabled, upon visiting one of their partners, a data subject can sign in through the SMP to avoid being tracked. However, this means that the SMP must inform the publisher of the visitor and that they have a subscription, allowing them to track which websites a user visits. *Contentpass* claims to only store the number of page views per publisher for billing reasons, as the SMP pays each website a small fee for each visit, and this information is erased after billing. However, the fact that data like this is stored even in the short term could mean that any data leaks on their databases can lead to users being tracked because of using SMPs, like *contentpass*.

5.2 CMPs and IDs

Among our results, we additionally found some CMPs track before consent, which may not be strictly related to any of our research questions. However, as the larger goal of this work is to improve security and user privacy, we believe it is important to present this finding for discussion.

When a data subject visits a website containing a CMP or SMP for the first time, they are assigned an identifier. These identifiers include a unique ID and a session ID, which are equivalent in this case as the purpose is to be able to track one session – a session can be infinitely long. An ID is assigned to the user to preserve the state of their choices to the CMP and to serve as proof of these choices. By assigning a list of consents to this ID, the controller knows what to do. The IDs are needed to enable tracking of data subjects in case they accept being tracked.

The fact that this ID is saved means that a user may be tracked and identified using it. Therefore, SMPs and CMPs can technically track users using information obtained before any given consent. From our results, we observed that some SMPs

and CMPs track interactions with the consent banner even when consent is not given. We also observed that some consent banners implement *lazy loading* – to load resources only when needed – that fetches resources when you interact with the consent banner. The *lazy loading* could be used to track data subjects. A CMP that did not exhibit signs of tracking data subjects' interaction with the consent banner is CookieYes as everything was loaded from the initial request. This means that *lazy loading* is not a strict requirement for the consent banner to function, and the use might allow for collecting data while still arguing that it is necessary to use *lazy loading* due to their implementation.

5.3 Consequences of CNAME cloaking

From the results, we identified that the partnered CMPs we tested with *contentpass* send requests to fetch *contentpass* code when loading the consent banner using CNAME cloaking. A small oversight in programming a publisher's website and not restricting the domains where associated cookies can be used can lead to significant privacy leaks. This is particularly concerning if users have sensitive data stored, such as personal information or secret tokens, that should not be accessible to other domains. In contrast to CMPs such as InMobi and CookieYes, their consent banner code is fetched as a third-party resource on a publisher's website, reducing the risk of first-party cookies leaking.

To support these findings. Conducting tests on HTML-tags (see Section 3.5, 3.6) and the use of CNAME cloaking, we can capture cookies stored in our browser that allow subdomains on server **B** when visiting server **A**.

5.4 Ethics and Lawfulness of SMPs

This section discusses the ethics and lawfulness of SMPs, including the purposes of CMPs and SMPs development, whether they fulfill their intended purpose, and the practical consequences of SMPs being implemented.

CMPs were developed as a consequence of the GDPR as a means to allow publishers to collect user data in a method that respects user privacy and consent. SMPs are the evolution of this idea, taking a step in a different direction. The potential issue with SMPs compared to CMPs is that, where CMPs allowed users to reject tracking free of charge – ideally making privacy a freely given right to users browsing the web –, this is not true for SMPs.

Contentpass claims on their website that they aim to "...offer privacy-conscious users & publishers a fair and simple alternative." [28] This raises the question of how to define the term *fair*. Can it be considered fair to demand a monthly subscription to the right of privacy from cookie tracking? And can this be legal, considering how CMPs offer the choice of rejecting cookie tracking for free under the GDPR?

As mentioned in 1.3, the GDPR states that consent is not considered freely given if a

user has no genuine or free choice to refuse. When the choice is to accept tracking or pay a fee, this may be considered not a freely given choice due to not being without detriment to the user. However, it is not fully clear whether this applies; after all, a user always has the option to not use the website as a means to deny consent. However, as shown by Mueller-Tribbensee et al. [24], the traffic to a website did not significantly reduce after implementing an SMP. This points towards users not utilizing this type of rejection in practice.

Concerning the ethics of SMPs, Mueller-Tribbensee et al. [24] provided some information regarding cookie paywalls which could explain why they are being used more and more, despite some of the questions regarding their legality. The authors found that pay or tracking walls (implemented by SMPs) are profitable, as they allow for more tracking than a normal cookie banner like a CMP. The authors also found that 99% of users accept tracking, probably due to the pricing being set so that a subscription is significantly more expensive than the data being collected is worth.

It is not within our scope to provide any firm conclusions as to what fairly given consent is, but we have found a good reason to question it. Mueller-Tribbensee et al. conclude their work by stating: "Pay-or-tracking walls seem to provide the means to expand the practice of tracking. Policymakers will need to consider whether this aligns with the goals of their legislation." [24] From the information we have found in combination with the related works, we have not discovered anything that contradicts this statement. Regardless of what the SMP claims their intentions are, the results of an SMP cookie banner being implemented on a publisher's website seem to be that it allows the publisher and the SMP themselves to profit by increasing user tracking through circumventing regulations like the GDPR.

5.5 Limitations

Here we include some of the limitations of our work.

There are two main areas in cookie management, the server side and the user side. A user will only use the API of the server with its CMP and will not touch the underlying mechanisms of how the cookies are generated, stored, or used. Our project will not go into detail about how cookie management is carried out in the web browser (user side) as it is application-specific. Handling of cookies might be susceptible to changes as cookies handling in an application is done programmatically and therefore is agreed by the industry of a default behavior to streamline the user experience. Thus we will only consider the server side since it is here we can do our research and findings to get an answer(s) to our questions.

Since the cookie market is vast and there can be multiple competing CMP/SMP, we have to limit ourselves to a few. Therefore, we only study the two most prevalent SMPs, *contentpass* and *Freechoice*. This research was also limited due to lack of access. Blackbox testing had to be performed on *Freechoice* and other unknown SMPs as we had no access to the software.

We also need to study the capabilities of CMPs. Hils et al. [11] conducted a study that measured the emergence of consent management by assessing the frequency of the various CMPs. From their findings, we will select the most prevalent CMPs.

The selection of only the most prevalent SMPs/CMPs may introduce bias in our findings, as this kind of selection can give us a limited view of what the real capabilities are. However, they are representative due to their popularity. Opting for a strategy of selecting services at random with replacement could be effective in reducing selection bias, but then the scope might be too large and harder to define.

As we were testing, we were experiencing problems with resources not loading on our website. We theorized that subdomains could be the reason for this. Either technical issues where subdomains could cause pathing issues or similar, or the tools given by CMP/SMP to test/implement a website might detect a subdomain and behave differently. We managed to get a domain to see whether it could solve our issues.

6

Conclusion

The goal of this thesis is to discover what SMPs are technically and legally under the GDPR, and how they relate to CMPs. This document describes our experimental methods to gather information about several different CMPs and SMPs and the results.

In this chapter, we will present the conclusion of our experiments. The answer to each of our research questions will be presented, to end with an answer to the high-level research question. We also mention future work the topic of this thesis.

6.1 Future work

Here we include some of the future work for the topic of this thesis.

One of the future works is to widen our research on SMPs by gaining access to more SMPs than just *contentpass*. This could include *Freechoice*, but also other lesser-known SMPs. Additionally, many CMPs have not been studied in this thesis. Among these are several CMPs which for instance partner with *contentpass* and may partner with other SMPs. To study SMPs it can be useful to study CMPs, with a focus on all the CMPs that partner with SMPs. Expanding this idea, studies can be made comparing CMPs to identify what qualities make a CMP eligible for partnering with an SMP, as this may be informative of which qualities the SMP demands for a functioning partnership.

A useful future work would be a guide or manual instructing research within SMPs and CMPs. This manual could consist of a step-by-step guide to creating a test environment and how to install CMPs and/or SMPs on a test website.

A different type of future work is a legal one, where SMPs can be identified and defined under the GDPR. It is not in our scope to do this, it is not even within our scope to say that the information we have found is sufficient for future work like this to be currently possible. More work can also be done on the legal side of SMPs, potentially to argue for new laws and regulations to come into place to control them if deemed necessary.

From our findings, we found a trend going towards using CNAME cloaking as a tool to circumvent a future where third-party resources are discouraged. As more

services might use this technique, we encourage future work on this topic for robust detection and prevention methods before it becomes prevalent.

We also pondered the consequences of *lazy loading* within CMPs and SMPs, but have not conducted any experiments on it. As *lazy loading* could lead to tracking, this technique is worth investigating under the scope of CMP and SMP implementations.

6.2 Research Questions

In this section we answer each of our research questions, to then answer the high-level research question of the thesis.

RQ1. What kind of data do SMPs collect, process, or handle? Can SMPs collect data for their own objectives?

A data subject is given two choices on the consent banner: 1) A user consents to tracking. The CMP becomes the controller and handles the consent signal with all options enabled. 2) A user subscribes. They must now submit sensitive information like email address, password, country of residence, and payment method to the SMP directly.

Additionally, the code from the SMP will always be loaded on a visit. From gaining an understanding of the implementation of an SMP, we have two cases: **i)** Black-box testing *Freechoice* SMP, the evidence points towards that the SMP behaves as a CMP with a payment mechanism. **ii)** Our practical testing with *contentpass* and black-box testing websites with unknown SMPs, points towards the SMP using a CMP, by adding the option of subscribing to the SMP to extend the functionality of the CMP with a payment mechanism. These SMPs also use CNAME cloaking.

When analyzing the cases. From case **i)** the SMP behaves as a CMP, they therefore can collect, process, and handle everything a CMP can do. From case **ii)** it is not initially clear what they can do, but due to CNAME cloaking, their subdomain is classified as first-party. Therefore the SMP receives all cookies where the domain-attribute allows subdomains to receive the same cookies. When a fetch to the SMP occurs, the browser will attach the cookies to the request that the SMP can collect.

Therefore, based on the choices and the cases, SMPs can collect as much or more information as CMPs. If a user subscribed to a specific SMP, then it is possible to track this user across multiple websites that implement this SMP.

RQ2. Do they have access to personal data? How?

If a data subject chooses to 1) accept all tracking, the SMP can use identifiers to track them, which can be used to extract personal data about a data subject. If a user chooses 2) and subscribes, some personal data must be voluntarily given to the SMP. Therefore, either through demanding information to be filled in upon a subscription activation or through tracking they will have access to some personal data.

RQ3. What is the relationship between SMPs and CMPs?

It seems to depend upon the implementation. As it looks from our black-box testing, the random sample of websites leaned towards using SMP to extend the functionality of CMP.

When an SMP is implemented as an extension of a CMP, their relationship is simple. The CMP handles the consent banner and if the data subject chooses to subscribe, then disable the consent banner.

The *Freechoice* SMP service seems to be one of the few where they build the payment mechanism to integrate into their CMP. In this case, the difference between the CMP and the SMP is less certain. The likely answer is that the SMP behaves as a CMP, where either the reject all or the accept all option is filled in for the data subject depending on their subscription status.

RQ4. What are the technical evidences in favor of SMPs being controllers and/or processors?
When an SMP extends a CMP, as is the case with contentpass, the SMP's only job is to receive payment and let the CMP handle the consent signal. Therefore it leans towards neither as it does not handle the consent signal directly, and therefore we found no technical evidences in favor of SMPs being controllers and/or processors.

RQ5. What is the relationship between SMPs and publishers?

When a publisher desires to import an SMP onto their website, they are pushed towards a form of partnership with the SMP. Once contact is established, the publishers will need to follow several steps to prepare their website for SMP integration, which includes utilizing a CMP of some kind. The publisher seems to have little influence over how the SMP is integrated, as presumably, the same steps need to be taken for each website wishing to partner with an SMP. This is somewhat different from many CMPs, where they allow the publisher some influence over the design and look of the cookie banner by offering a selection of different options to choose from.

Once the SMP is successfully integrated, no information flows between the SMP and the publisher. As mentioned, the SMP *contentpass* fits into the case **ii)**, either forcing the *accept all* if no subscription is activated or reject all if it is. Therefore we found no major difference between the relationship of SMPs and publishers compared to CMPs and publishers.

The high-level research question of this thesis is defined as:

What are SMPs, technically and legally, and how do they articulate with CMPs?

We found that there are different possible implementations of SMPs. As was the case with *contentpass*, where the SMP fits into the case **ii)**. In this case, the technical definition is an extended CMP and the legal definition is unclear, as the legal responsibility lands with the CMP. Alternatively, an SMP can be more closely

integrated with a CMP, as was found in *Freechoice*. This would make the technical definition very similar to a CMP as the SMP takes the role the CMP used previously. The legal definition would therefore likely be a processor of data. Whether or not this processor works within its limit of the scope of the controller and other more intimate details about the CMP relationship is not known as the research availability on *Freechoice* was limited. However, it is clear that SMPs and CMPs are very closely connected. This means that if a CMP stands on uncertain legal grounds, the SMP that the CMP is integrated into is highly likely to have the same issues.

In conclusion, SMPs can collect personal data through several means. Some questions should be raised regarding the method of data collection and the very nature of an SMP, as it raises concerns regarding the ethics of setting a price for privacy. SMPs also seem to be very closely connected to CMPs in different ways depending on the SMP, making the legality of CMPs crucial for SMPs. Therefore we would like to underscore the need for robust data protection measures and adherence to GDPR guidelines by both CMPs and SMPs to safeguard user information.

Bibliography

[1] L. Montulli and D. M. Kristol, "HTTP State Management Mechanism," Internet Engineering Task Force, Request for Comments RFC 2109, Feb. 1997, Num Pages: 21. DOI: 10.17487/RFC2109. [Online]. Available: https://datatracker.ietf.org/doc/rfc2109 (visited on 04/02/2024).

[2] European Parliament and Council of the European Union, "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)," *Official Journal of the European Union*, vol. 59, no. 1, pp. 1–88, 2016. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679.

[3] B. Wolford, *What are the gdpr fines?* [Online]. Available: https://gdpr.eu/fines/ (visited on 04/24/2024).

[4] C. Santos, M. Nouwens, M. Toth, N. Bielova, and V. Roca, "Consent management platforms under the gdpr: Processors and/or controllers?" In *Annual Privacy Forum*, Springer, 2021, pp. 47–69.

[5] European Data Protection Board, *European data protection board: Guidelines 07/2020 on the concepts of controller and processor in the gdpr version 1.0 (2020)*, 2020. [Online]. Available: https://edpb.europa.eu/our-work-tools/documents/public-consultations/2020/guidelines-072020-concepts-controller-and_en (visited on 05/22/2024).

[6] A. Rasaii, D. Gosain, and O. Gasser, "Thou shalt not reject: Analyzing accept-or-pay cookie banners on the web," in *Proceedings of the 2023 ACM on Internet Measurement Conference*, ser. IMC '23, Montreal QC, Canada, Association for Computing Machinery, 2023, pp. 154–161, ISBN: 9798400703829. DOI: 10.1145/3618257.3624846.

[7] European Data Protection Board, "Directive 2009/136/ec of the european parliament and of the council of 25 november 2009," 2002. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32009L0136.

[8] IAB Europe, *Transparency Consent Framework Policies*, 2020. [Online]. Available: https://iabeurope.eu/wp-content/uploads/2020/11/TCF_v2-0_Policy_version_2020-11-18-3.2a.docx-1.pdf.

[9] C. Santos, N. Bielova, and C. Matte, "Are cookie banners indeed compliant with the law? deciphering eu legal requirements on consent and technical means to verify compliance of cookie banners," *arXiv preprint arXiv:1912.07144*, 2019.

[10] C. Matte, N. Bielova, and C. Santos, "Do cookie banners respect my choice? : Measuring legal compliance of banners from iab europe's transparency and consent framework," in *2020 IEEE Symposium on Security and Privacy (SP)*, 2020, pp. 791–809. DOI: 10.1109/SP40000.2020.00076.

[11] M. Hils, D. W. Woods, and R. Böhme, "Measuring the emergence of consent management on the web," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 317–332.

[12] M. Degeling, C. Utz, C. Lentzsch, H. Hosseini, F. Schaub, and T. Holz, "We Value Your Privacy ... Now Take Some Cookies," en, *Informatik Spektrum*, vol. 42, no. 5, pp. 345–346, Oct. 2019, ISSN: 1432-122X. DOI: 10.1007/s00287-019-01201-1.

[13] Contentpass GmbH, *Contentpass website*, 2023. [Online]. Available: https://www.contentpass.net/ (visited on 01/2024).

[14] Traffective GmbH, *Freechoice website, 2023*. [Online]. Available: https://freechoice.club/ (visited on 01/2024).

[15] M. Nouwens, I. Liccardi, M. Veale, D. Karger, and L. Kagal, "Dark Patterns after the GDPR: Scraping Consent Pop-ups and Demonstrating their Influence," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20, New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 1–13, ISBN: 978-1-4503-6708-0. DOI: 10.1145/3313831.3376321.

[16] European Data Protection, *Guidelines 07/2020 on the concepts of controller and processor in the GDPR | European Data Protection Board*. [Online]. Available: https://www.edpb.europa.eu/our-work-tools/our-documents/guidelines/guidelines-072020-concepts-controller-and-processor-gdpr_en (visited on 05/22/2024).

[17] Consentmanager AB, *Working with contentpass integration*, 2023. [Online]. Available: https://help.consentmanager.net/books/cmp/page/working-with-contentpass-integration (visited on 01/2024).

[18] V. Morel, C. Santos, V. Fredholm, and A. Thunberg, "Legitimate interest is the new consent - large-scale measurement and legal compliance of iab europe tcf paywalls," in *Proceedings of the 22nd Workshop on Privacy in the Electronic Society*, ser. CCS '23, ACM, Nov. 2023. DOI: 10.1145/3603216.3624966.

[19] M. Toth, N. Bielova, and V. Roca, "On dark patterns and manipulation of website publishers by CMPs," en, *Proceedings on Privacy Enhancing Technologies (PoPETs)*, vol. 2022, no. 3, p. 478, 2022. DOI: 10.56553/popets-2022-0082. [Online]. Available: https://inria.hal.science/hal-03577024.

[20] A. Barth, "HTTP State Management Mechanism," Internet Engineering Task Force, Request for Comments RFC 6265, Apr. 2011, Num Pages: 37. DOI: 10.17487/RFC6265. [Online]. Available: https://datatracker.ietf.org/doc/rfc6265 (visited on 03/01/2024).

[21] J. R. Mayer and J. C. Mitchell, "Third-Party Web Tracking: Policy and Technology," in *2012 IEEE Symposium on Security and Privacy*, ISSN: 2375-

1207, May 2012, pp. 413–427. DOI: 10.1109/SP.2012.47. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6234427.

[22] H. Dao, J. Mazel, and K. Fukuda, "CNAME Cloaking-Based Tracking on the Web: Characterization, Detection, and Protection," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3873–3888, Sep. 2021, Conference Name: IEEE Transactions on Network and Service Management, ISSN: 1932-4537. DOI: 10.1109/TNSM.2021.3072874. [Online]. Available: https://ieeexplore.ieee.org/document/9403411.

[23] *What are DNS records?* en-us. [Online]. Available: https://www.cloudflare.com/learning/dns/dns-records/ (visited on 06/14/2024).

[24] T. Mueller-Tribbensee, K. M. Miller, and B. Skiera, *Paying for Privacy: Pay-or-Tracking Walls*, arXiv:2403.03610 [econ, q-fin], Mar. 2024. DOI: 10.48550/arXiv.2403.03610. [Online]. Available: http://arxiv.org/abs/2403.03610.

[25] *Noyb files GDPR complaint against Meta over "Pay or Okay"*, en. [Online]. Available: https://noyb.eu/en/noyb-files-gdpr-complaint-against-meta-over-pay-or-okay (visited on 06/17/2024).

[26] S. Nidhra, "Black Box and White Box Testing Techniques - A Literature Review," *International Journal of Embedded Systems and Applications*, vol. 2, pp. 29–50, Jun. 2012. DOI: 10.5121/ijesa.2012.2204.

[27] *Let's Encrypt*, en-US. [Online]. Available: https://letsencrypt.org/ (visited on 03/20/2024).

[28] Consentmanager AB, *Frequently asked questions*, 2024. [Online]. Available: https://www.contentpass.net/en/faq (visited on 01/2024).