Master Thesis
Computing Science

Radboud University

Post-Third-Party Cookies: Analyzing Google's Protected Audience API

Author:
Michiel Philipse
s1016359

Supervisor:
Güneş Acar
g.acar@cs.ru.nl

Second assessor:
Christine Utz
christine.utz@ru.nl

July 1, 2024

Abstract

In 2023, Google officially released the Privacy Sandbox: a collection of web technologies that are meant to eventually replace third-party cookies in Google Chrome. One of the APIs from the Privacy Sandbox is the Protected Audience API, which so far received little attention from the research community. This API facilitates ad auctions locally on the user's device, using ads suggested by websites previously visited by the user.

In this thesis, we analyze how this API is currently being used, based on data we got by crawling 10,000 popular websites. At the time of our crawl, Google's own DoubleClick was by far the most prevalent company using this new API. However, we also found six other advertising companies using the Protected Audience API. By looking at the provided function parameters that we collected during our crawl, we can see the suggested ads, how long the ads should remain, as well as whose ad auctions they may participate in and what auction logic they use. We did our crawl from both the US and the EU. When crawling from the EU, we got much fewer calls to join ad interest groups, likely due to EU regulations regarding user consent. While the Protected Audience API is a big improvement for the user's privacy compared to third-party cookies, many of its privacy protections are currently still quite weak to ease the transition for advertisers.

Acknowledgements

We thank Luqman Zagi for allowing us to use their link extractor, which we use in our crawler to collect inner links from websites in order to navigate to subpages. I also want to thank my supervisor Güneş Acar for all his valuable feedback and suggestions, and for supporting me throughout this project. His guidance, encouragement, and expertise were invaluable for this project.

Contents

1 Introduction

In the past few years, browsers have started taking more and more measures to protect the user against tracking. Firefox and Safari ship with built-in tracker protection [12, 82], and Brave blocks all trackers and ads by default [40]. However, the most widely used browser, Google Chrome, has not yet taken such measures. Google is itself one of the largest advertising companies, with its trackers on over 85% of popular websites [21]. But despite Google profiting from trackers by using the collected user data to deliver targeted advertisements, it appears they will now also finally follow the other browsers in a more privacy-friendly direction.

Google first announced it would soon start blocking third-party cookies in 2020 [11]. These third-party cookies are used by Google and other advertising companies to track users across different websites, which allows these companies to target advertisements toward individual users based on their browsing behavior. To replace some functionality that would be lost by blocking these third-party cookies, Google has developed the *Privacy Sandbox* [62]: a collection of browser APIs and other web technologies that aim to facilitate certain kinds of targeted advertising in a more privacy-friendly way.

Blocking third-party cookies could break existing functionalities on many sites. The advertising industry worries that they could lose revenue because of Google's decision to block third-party cookies, as not all functionalities that advertisers currently use are available with the Privacy Sandbox [3, 26]. As Google is itself one of the largest advertising companies, there are also concerns about anti-competitive behavior from Google. For this reason, the UK's Competition and Markets Authority (CMA) started an investigation into Google's Privacy Sandbox in 2021. While Google has been cooperating with CMA's investigation, the phaseout of third-party cookies in Chrome has now been delayed to at least 2025 in order to address any remaining concerns from the CMA [7, 51].

While the blocking of third-party cookies has been delayed, the new APIs from the Privacy Sandbox did become generally available in Chrome towards the end of 2023 [13]. As these APIs have only recently been released, it is interesting to investigate how they work and to see how they are being used. In this thesis, we will look into one of these new APIs that received little attention from the research community so far: the *Protected Audience API*.

The Protected Audience API allows advertisers on a website to add visiting users to interest groups. These interest groups contain ads, which can be shown later on when the user is browsing another website. For instance, if a user visits a product page on a web shop, the web shop can add the user to an interest group for that product to show ads to that user about that product at a later date. This kind of advertising is known as *remarketing* (also referred to as *retargeting*) [64], which is the main use case of the Protected Audience API. Traditionally, remarketing would use third-party cookies to recognize the user on another site. With the Protected Audience API, the interest groups and their associated ads are instead stored locally on the user's device. This allows the browser to display the ads later on without advertisers needing to track users across the web.

The main research question of this thesis is as follows:

> *How is the Protected Audience API currently being used by advertisers on popular websites during the initial release of the Privacy Sandbox?*

To answer this question, we have developed a crawler that collects data about the usage of the Protected Audience API. This crawler visits websites from a list of the 10,000 most popular websites to collect various data, including HTTP requests, redirects, and intercepted API calls. Based on this data, we will show which advertising companies are the early adopters of the API. By looking at the arguments they provide to the API calls, we will get an idea of how they use the API. By running the crawler from two different locations, we will also be able to compare how API usage in the US compares to usage in the EU. Furthermore, we will also look at how much the Protected Audience is used compared to the *Topics API*, which is one of the other APIs from the Privacy Sandbox used for targeted advertising.

In addition to the results from our crawl, which we will discuss in section 4, we will also discuss how the Protected Audience API aims to protect the user's privacy. There are multiple privacy measures in place, such as k-anonymity assurance and the usage of Trusted Execution Environments, but most of these measures are currently not enforced to their full extent yet, as we will discuss in section 5.1. We will touch on the topic of user consent in section 5.2, and how regulations like the GDPR might affect the usage of the Protected Audience API. In section 5.3, we will speculate on why Google has made the shift to more privacy-friendly advertising APIs with their Privacy Sandbox. But first, we will provide some more background information in section 2, about how targeted advertising used to work with third-party cookies, what Google's Privacy Sandbox consists of, and how exactly the Protected Audience API works.

2 Background

In this chapter, we discuss the concepts of real-time bidding and ad auctions, which are necessary to understand the workings of the Protected Audience API. We also provide some background on the APIs that make up Google's Privacy Sandbox, and we explain the fundamental workings of the Protected Audience API.

2.1 Real-time bidding & ad auctions

In the early days of the web, online advertisements worked similarly to advertisements in newspapers or on TV, where advertisers could buy an ad space directly from the website, newspaper, or TV network. However, unlike advertisements in newspapers or on TV, online advertisements are nowadays often highly personalized for the user who is viewing the ad. This allows online advertisers to target their specific target audience directly, which means advertisers spend less money on ad spaces for users who are less likely to be interested.

This personalization of online advertisements works by having advertisers bid on ad spaces in *ad auctions*. Since the personalization depends on the metadata of the user who will view the ad space, this bidding process must take place in real-time, which is why it is called *Real-Time Bidding* (RTB). The ad auctions usually go through an *ad exchange*, which connects publishers (those who sell ad spaces on their website or in their app) to multiple ad networks to find the most relevant ads [84].

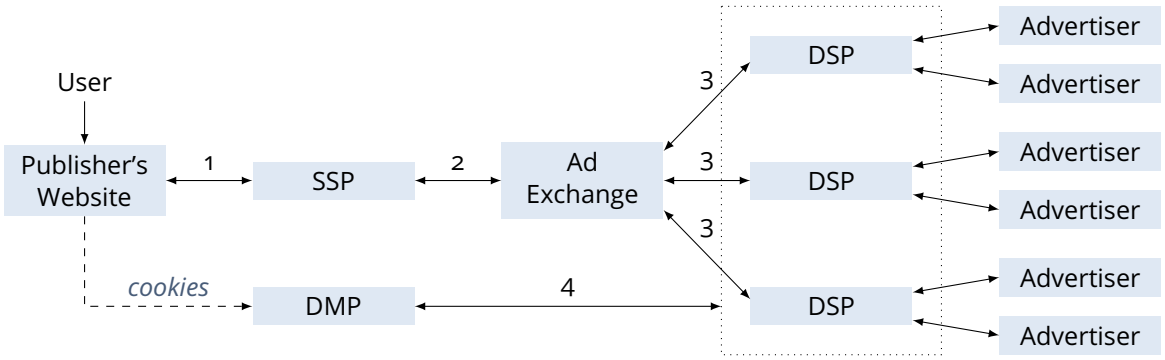

Figure 1: The interactions between the parties involved in real-time bidding for online ad auctions [85, 38]

As seen in figure 1, there are quite a few steps involved in these ad auctions:

1. Once a user visits the publisher's website, the website requests an ad from the *Supply-Side Platform* (SSP), which could be e.g. *Google Ad Manager*
2. The SSP connects to an *ad exchange*, e.g. *Google AdX* [38] (now part of Google Ad Manager), *Microsoft Advertising Exchange*, or *Amazon Publisher Services*
3. The ad exchange requests bids from one or more *Demand-Side Platforms* (DSPs), e.g. *Google Marketing Platform*, *Amazon DSP*, or *Yahoo DSP*
4. The DSPs generate bids for the various ad campaigns the advertisers are currently running, using user data from *Data Management Platforms* (DMPs) such as *Google Audience Center* (now part of Google Marketing Platform), which use third-party cookies to track user data for personalization

5. The DSPs send their best bid back to the ad exchange, and the ad exchange decides which of these bids wins the final auction, after which the winning ad is sent to the publisher's website to be displayed to the user

Note that the "best" bid is not always the highest bid. Ad exchanges and DSPs may introduce biases to optimize the number of clicks or conversations they get based on the advertiser [86]. Optimizing bids for RTB is an interesting field with lots of ongoing data science and machine learning research [42], but this is not the main focus of this thesis.

Third-party cookies play an important role in online ad actions because they allow a DMP to identify the user, giving the advertisers access to user data which can be used for the personalization of ads. The more you know about a user, the more value you can get out of an ad space by making the ads more relevant for that user. However, all this tracking done by advertising firms also leads to privacy concerns for the users who get tracked [19, 77].

Cookies can be used to store unique tracking identifiers, allowing websites to identify your browser when you come back to their site. *Third-party cookies* are cookies from third-party services, such as advertising or analytical services, and these cookies allow these services to track users across multiple different sites. Recent regulations such as the General Data Protection Regulation (GDPR) [25] have limited what companies are allowed to do with personally identifiable data without the user's consent. However, websites still find plenty of ways to track users, using e.g. browser fingerprinting, often even without their consent [43].

2.2 Google's Privacy Sandbox

In an attempt to limit cross-site tracking and to improve the privacy of the users, Google is working on a Privacy Sandbox: a collection of APIs that aim to replace third-party cookies, while still allowing advertisers to show relevant ads to users (which is certainly also very important for Google themselves). After some delays, Google started disabling third-party cookies for a small percentage of Chrome users in early 2024. The phaseout of third-party cookies has been pushed back multiple times due to concerns from the advertising industry, but they are currently aiming to disable third-party cookies completely in 2025 [51, 7].

The Privacy Sandbox has two main APIs that help advertisers show relevant content to users without third-party cookies: the *Topics API* and the *Protected Audience API*. The Protected Audience API [56] (previously known as *FLEDGE*) facilitates on-device ad auctions by having advertisers add users to interest groups. This API is the main subject of this thesis, and it will be discussed in more detail in section 2.3. The Topics API [61] allows advertisers to observe which general topics a user is interested in based on their recent browsing history. The Topics API replaces the *Federated Learning of Cohorts API* (FLoC), which was criticized for being opaque and facilitating cross-site user tracking [8]. However, despite Google's now more privacy-focussed approach, similar concerns still exist for the Topics API [10, 29, 5].

There are also several other supplementary APIs in the Privacy Sandbox. The *Attribution Reporting API* [44] and the *Private Aggregation API* [53] can be used to measure and report ad performance data such as conversions. The *Shared Storage API* [59] can be used by advertisers to store cross-site data, e.g. to show a sequence of ads in a certain order. The *Private State Token API* [54] can be used to distinguish real users from bots, by e.g. sharing the results of a CAPTCHA across multiple sites, and the *Federated Credential Management API* [47] can be used as a third-party login mechanism for websites, which previously often relied on third-party cookies.

In addition to these APIs, the Privacy Sandbox also includes some mechanisms to still facilitate third-party cookies across different sites under certain conditions. Companies that use multiple domains can use *Related Website Sets* [58] to indicate which websites belong to them, allowing cookies to be shared between these sites. For third-party cookies in embedded frames, *Cookies Having Independent Partitioned State* (CHIPS) [45] can be used, which are third-party cookies that are partitioned from the main site to prevent cross-site tracking.

While the advertising industry was afraid to lose a large part of its revenue from Google blocking third-party cookies in Chrome, the introduction of the Privacy Sandbox is expected to mitigate this loss in revenue for a large part [3]. However, as advertising is also a significant part of Google's business, this does bring some concerns for Google's monopoly position within the industry. These concerns are the main point of the CMA's investigation [7]. Thus far, it seems that Google is willing to address these concerns, although it did cause additional delays in the phaseout of third-party cookies [51].

2.3 Protected Audience API

In this thesis, we focus on the Protected Audience API. This API aims to provide a supposedly privacy-friendly way to do *remarketing* (sometimes also referred to as *retargeting*). Remarketing is a kind of advertising aimed at users who have previously visited the advertised website. For example, if a user looks at a product on a web shop, that web shop can then later advertise a discount for that product to that user. This is different from prospecting advertisements, which aim to get new users interested in a product or service [1].

In the past, remarketing worked by recognizing a user via third-party cookies set by the advertiser or their DSP. The Protected Audience API allows advertisers to add users to an *ad interest group* when a user visits the advertiser's website, indicating that the advertiser intends to show ads to this user in the near future. These interest groups are stored locally on the user's device for up to 30 days, which means that the advertiser no longer needs to track the user to show personalized ads. When another website wants to show the user an ad, the ad auction can be done locally on the user's device, with only a very limited amount of communication with external services.

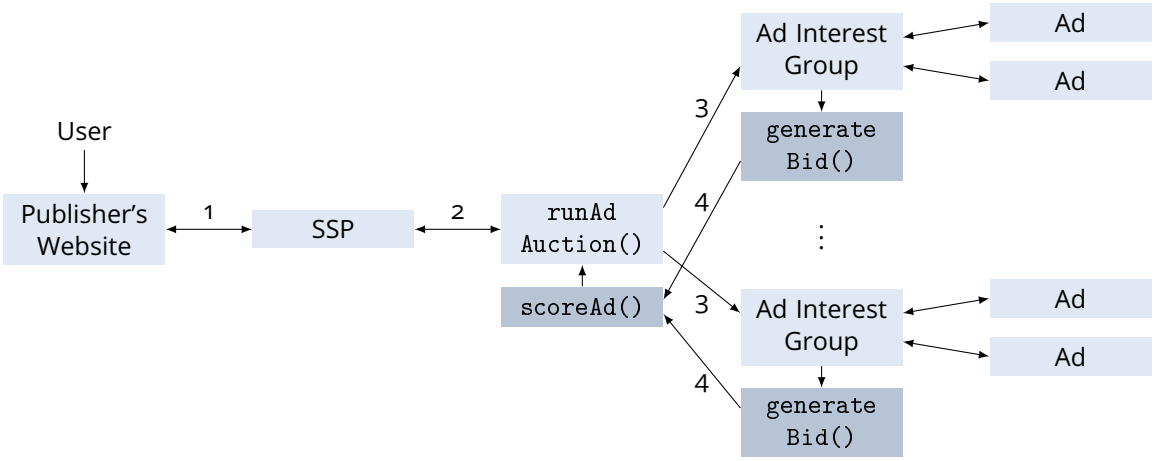

Figure 2: The interaction between the components involved in on-device real-time bidding using the Protected Audience API

As seen in figure 2, the infrastructure for on-device ad auctions with the Protected Audience API looks similar to the traditional ad auctions shown in figure 1. However, with the Protected Audience API, the ad auctions are done locally within the user's browser, using external auction logic scripts provided by the SSP and the DSPs:

1. Once a user visits the publisher's website, the website requests an ad from the SSP
2. The SSP runs `Navigator.runAdAuction` on the website with the appropriate parameters, which is a browser function implemented by the Protected Audience API
3. The browser then requests bids from one or more ad interest groups (managed by the DSPs), where every interest group links a *bidding logic script* containing a `generateBid` function, which generates a bid for an ad in the interest group based on the ad metadata and various auction signals
4. The browser then evaluates the bids from the interest groups using the `scoreAd` function from the *decision logic script* linked by the SSP, after which the highest-scoring ad wins the final auction and gets to be displayed on the website

Adding a user to an interest group is done by DSPs by calling `Navigator.joinAdInterestGroup`. Every interest group must have an owner (usually the DSP) and a group name. To partake in ad auctions, they should also have a list of ads, as well as a bidding logic URL linking to a script with a `generateBid` function. The data associated with an interest group may be updated by the owner via `Navigator.updateAdInterestGroups`, e.g. to change which ads might be shown. Interest groups can also be updated automatically once per day using an update URL. Finally, `Navigator.leaveAdInterestGroup` can be used to remove the user from an interest group.

To generate bids and score ads, the DSPs and SSPs want data to make an educated guess about the value of the auctioned ad space. This is why the DSPs and SSPs can provide various signals when calling `joinAdInterestGroup` or `runAdAuction`, allowing these signals to be used during the bidding and scoring process. For example, the user's topics from the Topics API could be provided as part of the `auctionSignals` provided to `runAdAuction`, or as part of the `userBiddingSignals` provided to `joinAdInterestGroup` [56]. In addition to these provided signals, the DSPs and SSPs can also request "trusted signals" from *key/value services*. These trusted signals are meant to be used for frequently updated data, such as the amount of budget that is left in a certain advertiser's ad campaign. We will discuss these key/value services in-depth in section 5.1.1.

3 Related work

While the initial proposals for Google's Privacy Sandbox were first introduced in 2019 [67], APIs like the Protected Audience API and the Topics API have only recently become available for testing. Because of this, not much scientific literature has been published about the Privacy Sandbox APIs so far. Here we discuss the related work that has been published about the Privacy Sandbox, as well as some recent works regarding alternative proposals for privacy-friendly advertising.

3.1 About Google's Privacy Sandbox

In Sept. 2022, RTB House published a paper about experiments they did with an early trial version of the Protected Audience API [65]. RTB House is a Polish advertising company that focuses on retargeting. During their 5-month trial, they reportedly added around 1.2 million users to interest groups and they got over 7 million impressions from ads from these interest groups. However, the trial version of the Protected Audience API that they used was still missing many features. For example, there was no k-anonymity protection to make sure that ad data could not be used to identify individual users, which we will discuss more in section 5.1.2. Furthermore, only a very small percentage of users took part in the trial. Because of this, they do not yet have definitive results as to how this new API compares to traditional ad auctions using third-party cookies.

Google researchers have also published papers about some of the Privacy Sandbox APIs. Aksu et al. discussed how summary reports from the Attribution Reporting API can best be optimized, such that advertisers can get the maximum utility out of the reports despite the limitations put in place to protect the users' privacy [2]. Lachner et al. surveyed what browser controls would be liked most by users from a privacy point-of-view for inferred browsing topics, the results of which were used in the development of the Topics API [32].

More recently, several papers have been written about possible re-identification attacks with the Topics API [10, 29, 5]. These papers aim to determine the likelihood of a website being able to re-identify a user based on the user's interest topics from the Topics API. The Topics API adds noise to the topics it returns to make it harder to identify users. However, with enough data from multiple calls to the Topics API, this noise can be filtered out and users can often still be re-identified. Even with just a single Topics API call, Beugin and McDaniel found that based on a dataset of browsing histories from real users, 46% of users could be uniquely re-identified based on their interest topics [9].

The investigation from the UK's Competition and Markets Authority regarding Google's Privacy Sandbox [7] caused delays in the phaseout of third-party cookies in Chrome [51]. However, this investigation focuses on the concerns of advertising companies, not the user's privacy. As Nottingham points out [39], other browsers like Firefox have been blocking trackers for a while, so Google's decision to block third-party cookies should not be so controversial. Instead, Nottingham says the CMA should be more worried about unilateral behavior: decisions made by Google that differ from other browsers, such as the implementation of the new Privacy Sandbox APIs.

Google has been fairly open with the development of the Privacy Sandbox. Discussions about the API took place on GitHub, and they involved the W3C's Private Advertising Technology Community Group. This openness helps keep a good balance between the user's privacy and the demands of the advertising industry. Olejnik [41] discusses how the Privacy Sandbox could best be governed to ensure that this balance remains. They suggest that an independent entity should govern the Privacy Sandbox, with a special focus on the user's privacy. This could work similarly to other W3C working groups, which manage various other web API proposals. However, as Olejnik points out, the W3C working groups typically do not focus on issues regarding fair competition.

Johnson et al. pose various research questions that the advertising industry will need to look into in order to effectively make the switch to more privacy-friendly advertising [30]. The IAB Tech Lab provides a more concrete list of functionalities that advertisers might miss out on when they are forced to switch from third-party cookies to using the Privacy Sandbox [26]. Common use cases that are not supported by the Privacy Sandbox include for example *look-alike modeling* (used to target users who behave similarly to a brand's existing audience) and *exclusion targeting* (used to e.g. not bid on ad spaces from users who have already visited your site in the past).

Eliot and Murakami Wood [22] seem to suggest that Google is moving away from advertising, in favor of focusing primarily on AI. While Google does a lot of research regarding AI, we do not believe that they are moving away from advertising. Rather, the Privacy Sandbox seems to indicate that they still care very much about advertising. We think the Privacy Sandbox will only help make Google's targeted advertising business more sustainable, as we will discuss in section 5.3.

3.2 About Privacy Sandbox alternatives

Google is not the only company focusing on more privacy-friendly targeted advertising solutions. In the past few years, various new proposals have been published for advertising APIs to replace third-party cookies. These proposals come from large companies like Microsoft, Apple, Mozilla and Meta, as well as other advertising companies. Here we will discuss some papers published about these new proposals to illustrate how they compare to Google's Privacy Sandbox.

RTB House has published a paper about frequency capping, which aims to limit the number of times a user sees a particular advertisement within a certain time frame [27]. They used to use third-party cookies for this, but in this paper, they investigate alternative methods of implementing frequency capping without third-party cookies. With the Protected Audience API, browser signals can be used for frequency capping, or they can use the Shared Storage API. They also discuss PARAKEET [37], a proposal from Microsoft also based on TURTLEDOVE [80]. Unlike the Protected Audience API, which implements the TURTLEDOVE proposal more closely, Microsoft's PARAKEET does not do on-device ad auctions. Instead, it does the ad auctions in Trusted Execution Environments, similar to the key/value services we will discuss in section 5.1.1. However, for frequency capping with PARAKEET, the same browser signals can be used, just like with the Protected Audience API.

Researchers at Mozilla have written critical papers about various proposals. The *SWAN* [68] and *Unified ID 2.0* [76] proposals both allow advertisers to assign an identifier to a user for tracking purposes. They do ask the user for consent, and they have policies in place for how data may be used. However, Thomson and Rescorla say that both of these proposals facilitate the tracking of users in a way that bypasses the browser's anti-tracking protection [74]. Thomson also did a critical analysis of Apple's *Private Click Measurement* (PCM) [83], which is used to track clicks on advertisements and subsequent conversions, similar to Google's Attribution Reporting API. According to Thomson, the limited number of available identifiers in PCM encourages sites to aggregate events, but it does not stop malicious sites from tracking a limited amount of users [72].

Together with Meta, Mozilla has developed an attribution proposal of their own, called *Interoperable Private Attribution* (IPA) [73]. Similar to PCM and the Attribution Reporting API, IPA can also be used to track conversions from ads, but IPA does this by using multi-party computation to keep data private. McGuigan et al. have compared these three private attribution APIs from Meta and Mozilla, Google, and Apple [36]. They show how these companies have differing definitions of what privacy means in their online documentation and articles. Overall, these companies focus mainly on the user's anonymity or on limiting access to personal data when discussing privacy. McGuigan et al. argue that privacy should be more about the ethical implications of the processing of the user's data in certain contexts. Similarly, Martin et al. also point out that how exactly data is processed matters not nearly as much as whether the purpose of the processing is contextually appropriate [35].

According to a survey by Jerath and Miller [28], some privacy-enhancing technologies improve the user's perceived privacy much more than other technologies. The survey of 1,751 US users showed that group-level targeting (e.g. based on the user's interests) does very little to improve the user's perceived privacy on its own. However, keeping data on the user's device (e.g. using on-device ad auctions from the Protected Audience API) affects the perceived privacy much more. A different survey by Cooper et al. [19] found that most people prefer relevant ads, although many have privacy concerns about companies collecting their data. Overall, they show that Google's Privacy Sandbox seems to be better perceived by users compared to the Unified ID 2.0 proposal.

4 Usage measurements

In this chapter, we will explain how we built a crawler to collect data about how the Protected Audience API is being used on the web. We will also discuss the results we got from crawling 10,000 sites with this crawler, and we show some differences between the results we got while crawling from the US and the EU.

4.1 Our crawler

In order to gain insight into how and how much the Protected Audience API is currently being used, we will crawl the internet and collect all calls to methods from the Protected Audience API. To do this, we have modified DuckDuckGo's Tracker Radar Collector [20] to collect these calls[1]. Tracker Radar Collector uses Puppeteer to control an instance of Chrome, which allows it to interact with websites and gather data on the web. To ensure that the Privacy Sandbox APIs are enabled for our crawler, we set various Chromium flags as described in the documentation [56, 61]. Figure 3 summarizes the overall structure of our crawler.

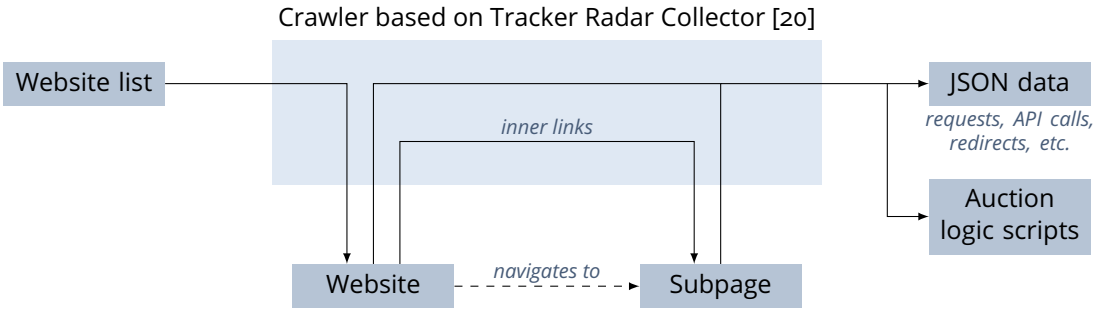

Figure 3: Summary of the crawler pipeline showing how data is collected

JSON Data Our crawler stores its results per crawled website in a JSON file. These JSON files, as shown in figure 4, contain data from various *collectors*. The *API call collector* collects all Protected Audience API calls to `joinAdInterestGroup`, `updateAdInterestGroups`, `leaveAdInterestGroup`, and `runAdAuction` on the `Navigator` interface. We also collect calls to `Document.browsingTopics` from the Topics API for comparison. Our API call collector is based on a script by Senol and Acar [69], which intercepts browser API calls to collect function arguments and metadata. It does this by overwriting the API methods with wrapper functions that extract the data before calling the original method. The API call collector is also responsible for interacting with the page. By scrolling to the bottom of the page and back up a few seconds after the page has loaded, we make sure that lazy-loaded elements get activated. The other collectors we used during our crawl are the *request collector*, which collects all HTTP requests during the crawl, and the *CMP collector*, which tries to automatically give consent when it detects a consent dialog.

[1] Our modified version of DuckDuckGo's Tracker Radar Collector is available on GitHub at `https://github.com/MichielP1807/tracker-radar-collector/`

asics.com_706b.json


```
{
  "initialUrl": "http://asics.com/",
  "finalUrl": "https://www.asics.com/nl/nl-nl/",
  "timeout": false,
  "testStarted": 1701807539704,
  "testFinished": 1701807641536,
  "data": {
    "requests": [...],
    "cmps": [{"name": "Onetrust", "succeeded": true, ...}],
    "privacySandbox": {
      "callStats": {
        "https://td.doubleclick.net/td/rul/952563132?random=17018...": {
          "Navigator.joinAdInterestGroup": 1
        },
        "https://fledge.teads.tv/v1/interest-group/tag.html": {
          "Navigator.joinAdInterestGroup": 46
        }, ...
      },
      "crawledSubpages": [{"initialUrl": "https://www.asics....", ...}],
      "savedCalls": [{
        "source": "https://td.doubleclick.net/td/rul/952563132?rand...",
        "description": "Navigator.joinAdInterestGroup",
        "arguments": {
          "0": {
            "owner": "https://td.doubleclick.net",
            "name": "1j825792954",
            "biddingLogicUrl": "https://td.doubleclick.net/td/bjs",
            "ads": [{"renderUrl": "https://tdsf.doublec...", ...}, ...],
            ...
          },
          "1": 2592000
        },
      ...
}
```


(the full file was 4.4MB in total)

Figure 4: A (shortened) example of the JSON data our crawler produced for `asics.com`, showing that multiple calls to `joinAdInterestGroup` were collected on this site

Logic scripts In addition to these JSON files, we also collect bidding and decision logic scripts during our crawl. We modified the API call collector to download these scripts automatically from the URLs provided in calls to `joinAdInterestGroup` and `runAdAuction`. These scripts are then saved in a separate folder per crawled site for later analysis.

Website list For our main crawl for this thesis, we use a Tranco list [33] generated on the 3rd of December 2023[2], which consists of the 10,000 most used websites that are included in the Chrome User Experience Report (CrUX) of October 2023. We limit our list to only sites included in the Chrome User Experience Report in an attempt to reduce the number of content delivery networks (CDNs), DNS servers, and similar services in our list. In early testing, we saw that these sites usually do not give useful results, and often do not connect as they do not necessarily have a home page. However, despite limiting our list to sites in the CrUX, the list still contained plenty of these services.

Subpages In addition to crawling the main page of the websites in our Tranco list, we also experimented with crawling subpages of the websites we crawl. We get the subpages by extracting the links on the main page. We then pick the links closest to the center of the screen, but only if they link to a different page on the same domain (so-called *inner links*). We did some test runs of our crawler with 1,000 sites, crawling two additional subpages on each site. As described in appendix A, we found that the second subpage gave very few new results. This is why we decided for our final 10,000-site crawl to only crawl one subpage per site, as crawling more subpages is not worth the extra time spent on each website.

[2]The Tranco list is available at `https://tranco-list.eu/list/N7VQW/10000`

4.2 Crawl results

On the 4th of December 2023, we started our crawl on both a DigitalOcean server in New York and a DigitalOcean server in Amsterdam. Over the course of 3 days, both servers crawled the 10,000 websites from our Tranco list. Figure 5 shows the general results from these crawls: 7,849 websites were successfully crawled from within the EU, and 7,209 websites were successfully crawled from within the US. The remaining 17% or 24% of websites either did not connect or caused some browser error while crawling (for example ending up on `chrome-error://chromewebdata/`).

Around 400 of the 10,000 domains we crawled ended up being duplicates. These are sites that ended up on the same (sub-)domain as at least one other website from our Tranco list based on their first-level domains (FLDs). For example, the most common final FLD for our crawl was `google.com`, to which 21 different domains redirected, including sites such as `gmail.com` and `doubleclick.net`, which means that 20 of these sites are shown as duplicates in figure 5.

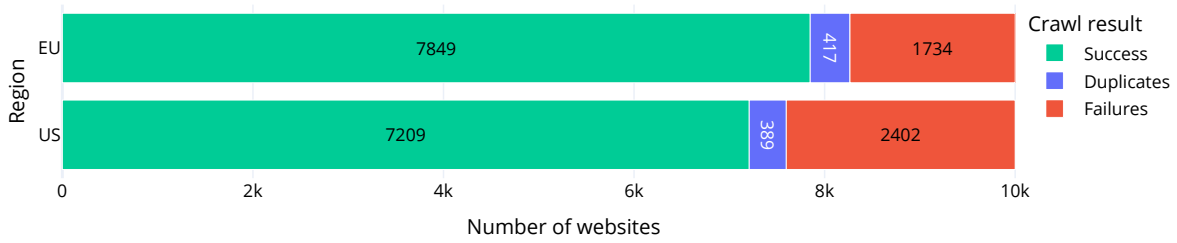

Figure 5: The results of our 10,000-site crawl from both the US and the EU

In total, our crawler collected 14,221 API calls in the EU and 12,690 API calls in the US. Figure 6 shows the number of API calls we collected per API method, and figure 7 shows the number of distinct websites on which the API methods were called. There is no data for the `updateAdInterestGroups` method, as this method was called on none of the crawled sites that we visited during our crawls. This is likely because this method is not the only way to update an interest group, as we will discuss in section 4.2.1.

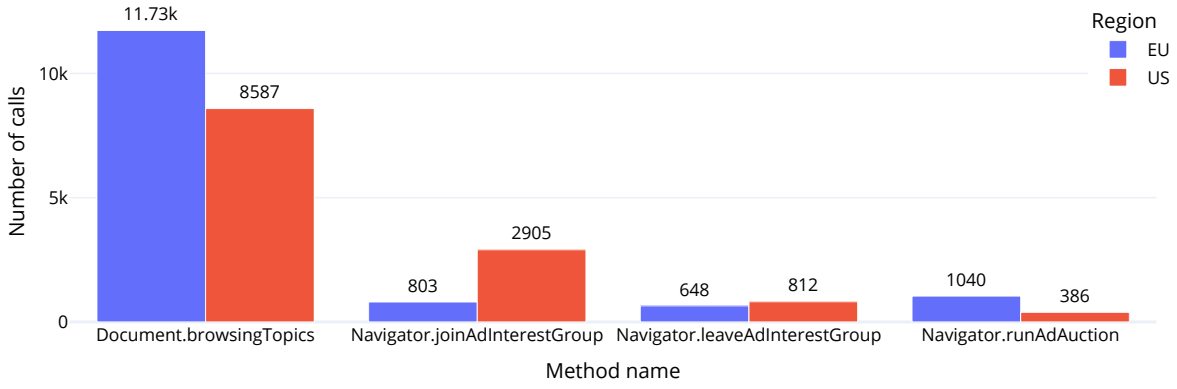

Figure 6: The total number of API calls collected per API method

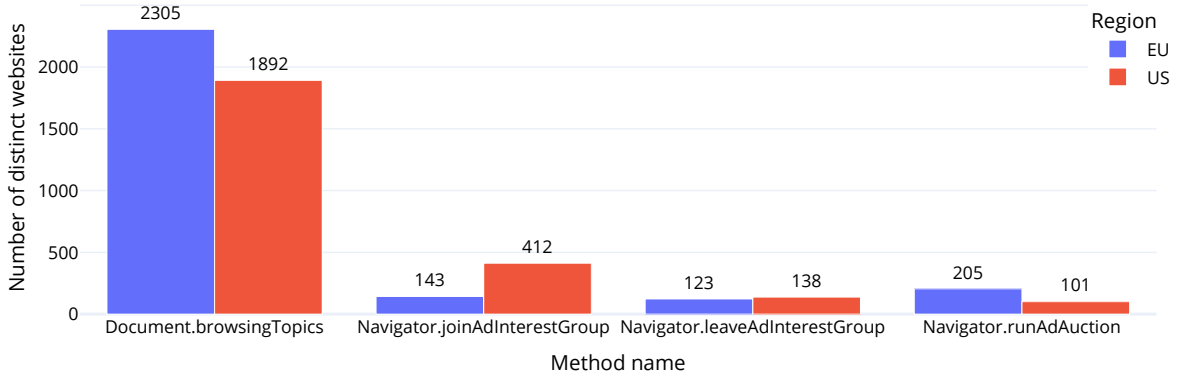

Figure 7: The number of distinct websites (based on their FLD) per API method, on which that method was called at least once

Protected Audience API Of the successfully crawled websites from both crawls, we found that on around 5.58% of these `joinAdInterestGroup` was called at least once, and on 3.44% `runAdAuction` was called at least once. Especially for `runAdAuction`, this is much more frequent compared to what the Chrome Platform Status reports. On the 1st of December, the Chrome Platform Status reports `joinAdInterestGroup` getting called on 3.55% of page loads [15], and `runAdAuction` getting called on only 0.66% of page loads [16]. This difference is likely caused by the fact that we counted the percentage of successfully crawled sites that used one of these methods, while the Chrome Platform Status reports the percentage of page loads that use a feature. This means that the data from the Chrome Platform Status is influenced much more by the most popular websites that are visited more often by users.

Interestingly, while we had more successfully crawled sites in our EU crawl than in our US crawl, we found that `joinAdInterestGroup` was called on almost three times as many websites from within the US compared to the EU (412 vs. 143). We speculate that this could be because of concerns about privacy regulations regarding user consent, as the EU tends to be stricter in this regard, which we will discuss more in section 5.2. On the other hand, `runAdAuction` was called on more than twice as many sites in the EU compared to the US (205 vs. 101). However, websites and advertisers are not always consistent in calling these functions every time the page is loaded. Considering the relatively small number of sites currently using the Protected Audience API, this difference might be somewhat negligible.

Topics API In figure 7, we can see that the Topics API's `browsingTopics` method is used on many more websites compared to the Protected Audience API (2,572 vs. 809 sites for both crawls combined). This might be because the Topics API is easier to use. The Topics API consists of just one function that can be called to get the user's topics, while using the Protected Audience API requires more infrastructure to be set up by the SSPs and DSPs. Furthermore, the Protected Audience API is specifically for remarketing, while the Topics API has broader applications. Advertisers using the Protected Audience API may also want to use the user's topics from the Topics API as a signal to use during bidding [56]. Of the 276 total websites on which `runAdAuction` was called during our crawl, 223 also called `browsingTopics`, which indicates the topics might be used for bidding. However, many advertisers may use the Topics API for general advertising without the Protected Audience API.

Who calls these APIs? By looking at the stack trace of the intercepted API calls, we can see where the scripts that call the API methods come from. Figure 8 shows the most common source domains per API method, based on the number of distinct websites they call that method on. For calls to `browsingTopics`, we only show the ten most prevalent domains, as we found 65 domains using it in total, most of which appeared only on a few sites.

The top callers all seem to be advertising companies. For the Protected Audience API, by far the most prevalent domain is `doubleclick.net`, which is Google's own advertising platform. As these APIs are part of Google's Privacy Sandbox, it makes sense that Google is one of the first to start using these APIs themselves. Other than Google, we also found *Seedtag*, a DSP and SSP provider focused on contextual advertising, using both `joinAdInterestGroup` and `runAdAuction`. Furthermore, we found the DSP and SSP providers *Teads* and *MicroAd*, as well as the DSPs *Logicad* (`ladsp.com`) and *RTB House* (`creativecdn.com`) using `joinAdInterestGroup`. Interestingly, we found the DSP/SSP provider *Retargetly* using `leaveAdInterestGroup`, even though they did not call `joinAdInterestGroup` during our crawl.

During early experiments with our crawler, we also saw the French DSP and SSP provider *Criteo* doing some tests with the Protected Audience API. However, in our final crawl, we only got Topics API calls from them. Generally, the Topics API was called by many more companies, including e.g. LinkedIn (`licdn.com`), Magnite (`rubiconproject.com`), and Taboola. Interestingly, DoubleClick called the Topics API more than twice as often in our EU crawl compared to the US (1047 vs. 500).

In the remainder of this chapter, we will further analyze how the Protected Audience API is being used by these companies. By analyzing the collected parameters from the intercepted API calls, we will get some insight into how they use ad interest groups and ad auctions.

4.2.1 Ad interest group data

Each interest group has an owner and a name. The owner corresponds to the source that is calling the API, as scripts can only join, update, or leave interest groups that they own. The interest group names can be any string provided by the owner. While some owners may use readable names for their interest groups, e.g. to represent the site from which the interest group originates, most owners seem to use a random-looking identifier, as seen in table 1.

Owner	#	Interest group names
`https://td.doubleclick.net`	1383	`1j7122563249`, `1j7485685777`, `1j7384999739`, ...
`https://fledge.ladsp.com`	9	`113536`, `60748`, `60747`, `60746`, `60745`, `111953`, `111952`, ...
`https://fledge.teads.tv`	5	`ouraring.com`, `fairmont.com`, `kia.com`, `asics.com`, ...
`https://privacy-sandbox-ot.send.microad.jp`	3	`1:ad907019656ce2f8115571a39beb189fedcddc07a0c3515e`, `2:44042e85-6f4b-4895-ae84-3679c306c222`, ...
`https://t.seedtag.com`	2	`m.marca.com`, `parlons-basket.com`
`https://f.creativecdn.com`	1	`3QZNG4Cy4LGjFGhNZ6fL`
`https://track.u.send.microad.jp`	1	`1524:087e196d-ce5c-4f7b-b2e1-0cdee01dd222`

Table 1: The number of unique interest group names per owner with some example names from our collected `joinAdInterestGroup` calls

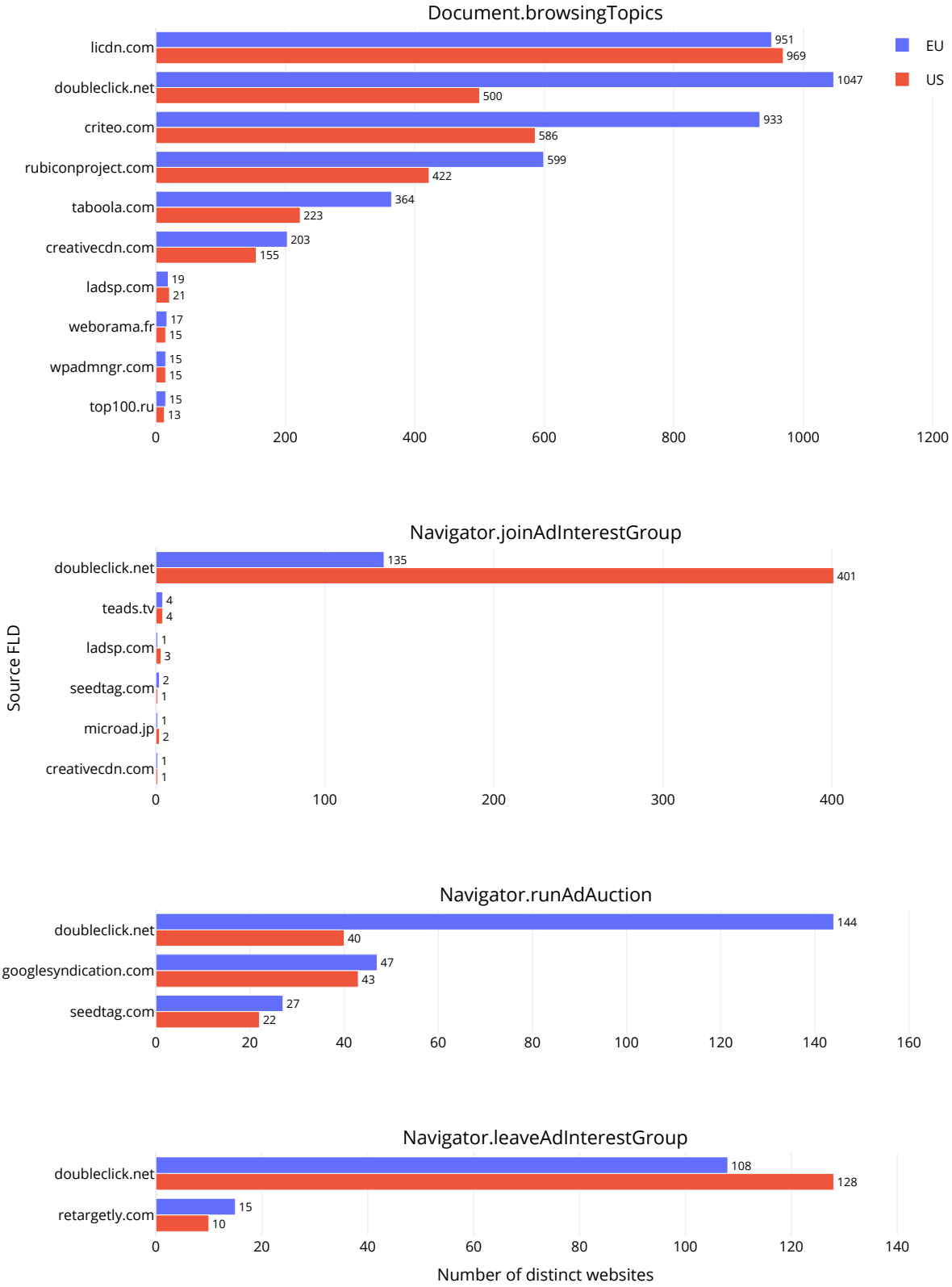

Figure 8: The number of distinct websites on which each source domain called the listed API method at least once (limited to the ten most prevalent domains for `browsingTopics`)

Durations The *duration* or *lifetime* of an interest group determines how long an interest group can participate in ad auctions. After this duration, the group will be removed from the user's device. According to the documentation [56], the duration of an interest group is capped at 30 days. However, many interest groups owned by `doubleclick.net` seem to try to set a much longer duration, as seen in figure 9. The longest interest group durations were provided by `creativecdn.com`, who seem to have entered 30 days in milliseconds instead of seconds, causing their interest groups to want to use a duration of 30,000 days. In the end, this does not matter much, as these durations should be limited to a maximum of 30 days by the browser in any case.

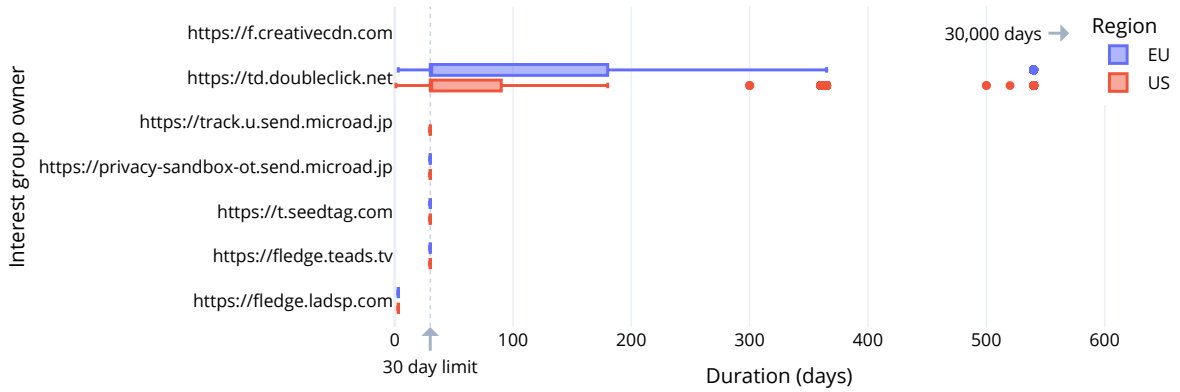

Figure 9: The durations of the interest groups per owner, where the 4 interest groups from `creativecdn.com` all had a duration of 30,000 days, which falls outside the range of this plot

Ads In order to participate in ad auctions, an interest group must have one or more ads associated with it. We found that interest groups contained around 25 ads on average, although many groups contained well over 100 ads, as seen in figure 10. These large groups seem to contain many variations of similar ads with e.g. differing layouts and differing sizes, such as the ones shown in figure 11.

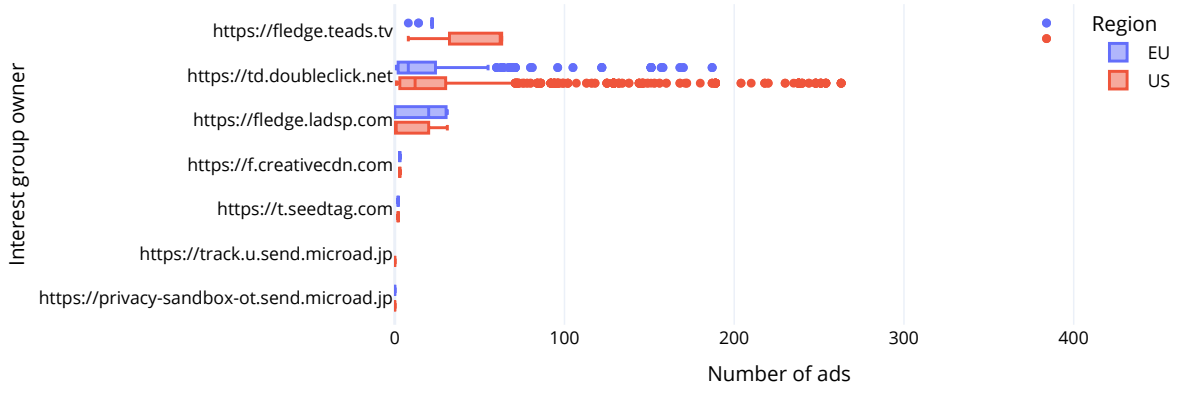

Figure 10: The number of ads in an interest group per owner

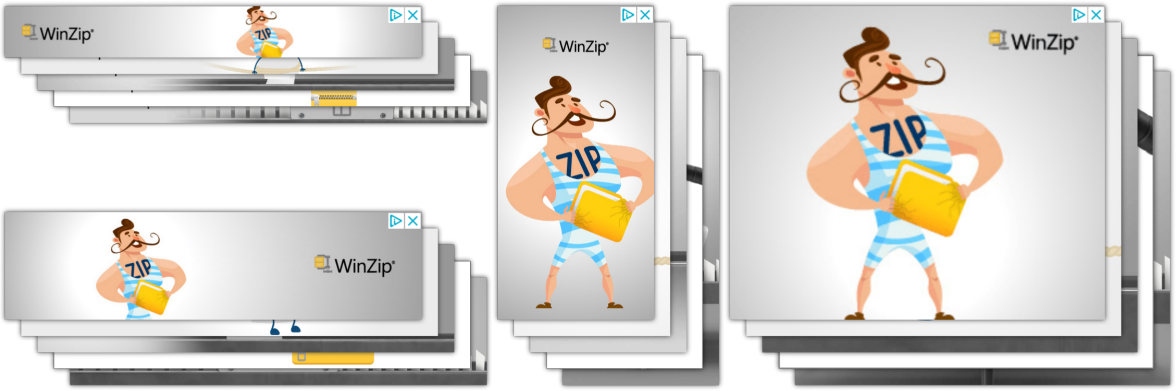

Figure 11: Some ads from `winzip.com`, showing a few ad variants in multiple sizes, all from the same interest group (this group contained 263 ads in total)

As an alternative to the standard ads for interest groups, advertisers can also use *ad components* in their interest groups. Ad components are used for ads that consist of multiple parts, e.g. to showcase multiple products from a web shop within the same ad. During our crawl, we only saw `creativecdn.com` using ad components on `rtbhouse.com`, with 65 ad components in each of their interest groups. As `rtbhouse.com` is the advertising company that operates `creativecdn.com`, it seems that they are only testing the ad components on their own website at the moment of this crawl.

Updates Some interest groups did not contain any ads when our crawler joined them, but these interest groups could be updated to contain ads in the future. During our crawls, we did not come across any calls to `updateAdInterestGroups`. This is in line with the Chrome Platform Status, which shows less than 0.002% of page loads calling this method at the time of our crawl [17]. However, this is not the only way interest groups can be updated. If the interest group contains a daily update URL, then the browser will automatically update the group once per day after an ad auction. Of all the interest group owners, only `doubleclick.net`, `teads.tv`, and `microad.jp` included daily update URLs in their interest groups. Another owner, `ladsp.com`, did not include a daily update URL but did include an `updateURL` property, which seems to be the new name for the `dailyUpdateUrl` property [46]. The owners use the same update URL for all of their groups, although `doubleclick.net` and `ladsp.com` do include some URL parameters with e.g. the group name in their update URLs, as seen in table 2.

Owner	Daily update URL (URL parameters may vary)
`https://td.doubleclick.net`	`https://td.doubleclick.net/td/update?ig_name=1j100169791&tag_eid=44800267`
`https://fledge.teads.tv`	`https://fledge.teads.tv/v1/interest-group/update`
`https://privacy-sandbox-ot.send.microad.jp`	`https://privacy-sandbox-ot.send.microad.jp/igapi/u?atids=`
`https://fledge.ladsp.com`	`https://fledge.ladsp.com/update_ig?ig=113536` (listed as `updateURL` instead of `dailyUpdateUrl`)

Table 2: The daily update URLs per owner from our collected `joinAdInterestGroup` calls

4.2.2 Ad auction data

Ad auctions are run by sellers, and these sellers decide which interest group owners may participate in their ad auctions. This is specified by the `interestGroupBuyers` property, which contains a list of buyers that are allowed to participate. While three different script domains called `runAdAuction`, we only found two different auction sellers: `doubleclick.net` and `seedtag.com`. This is because `googlesyndication.com` is also owned by Google, and internally also shows `doubleclick.net` as the seller. Table 3 lists the interest group owners to whom the sellers may sell their ad spaces. Which of these owners may participate depends on the particular ad auction. Interestingly, both `doubleclick.net` and `seedtag.com` ran some ad auctions during our crawl in which `creativecdn.com` was the only buyer.

Seller	Interest group owners (buyers)
`https://securepubads.g.doubleclick.net`	`https://td.doubleclick.net`, `https://googleads.g.doubleclick.net`, `https://f.creativecdn.com`
`https://t.seedtag.com`	`https://t.seedtag.com`, `https://f.creativecdn.com`

Table 3: All buyers that may bid on ad spaces in ad auctions from the sellers (although in some auctions, only a subset of the buyers may be allowed to participate)

4.2.3 Auction logic scripts

As explained in section 2.3, ad interest groups must link a bidding logic script to generate a bid for their ads, and ad auctions must link a decision logic script to score these bids. During our crawl, whenever we came across a call to `joinAdInterestGroup` or `runAdAuction`, we had our crawler automatically download the linked logic script from the URL included in the interest group or ad auction data. This resulted in 3,708 bidding logic scripts, and 1,426 decision logic scripts getting downloaded in total.

Interest group owners and auction sellers always seem to use the same bidding or decision logic URL. However, this URL did not always return the same script. From DoubleClick, we got 13 different bidding scripts and 5 different decision logic scripts, as seen in tables 4 and 5. Of the other sources, `creativecdn.com` and `teads.tv` also have at least two different bidding logic scripts. Since these different scripts came from the same URL, it seems like they might be testing various bidding strategies to see which performs best.

The logic scripts themselves are often hard to read, as most of them are long and minified. Only some scripts from `microad.jp` and `seedtag.com` were not minified, but these were very basic scripts without much interesting logic. Figures 12 and 13 show the sizes of the bidding and decision logic scripts per source. DoubleClick uses by far the largest scripts of all the sources we found using the Protected Audience API. This is at least partly because their bidding scripts contain large arrays of numerical values, which look like they could potentially be weights and biases for a neural network, or other vector data.

In addition to these logic scripts, the Protected Audience API also gives interest group owners the option to link a WebAssembly binary that can be used to help the browser compute bids faster. We did not analyze these WebAssembly binaries within our crawl, but from our interest group data, we can see that `doubleclick.net`, `creativecdn.com`, and `ladsp.com` are making use of this feature.

Owner	Bidding logic script URL	Unique scripts
doubleclick.net	`td.doubleclick.net/td/bjs`	13
creativecdn.com	`f.creativecdn.com/statics/buyer.js`	2
teads.tv	`fledge.teads.tv/v1/bidding/bidding-logic.js`	2
ladsp.com	`fledge.ladsp.com/bidding_logic/v0/bidding_logic.js`	1
microad.jp	`privacy-sandbox-ot.send.microad.jp/static/scripts/biddingLogic.min.js`	1
	`track.u.send.microad.jp/paa/v1/biddinglogic.js`	1
seedtag.com	`t.seedtag.com/ps/buyer/bidding-logic.js`	1

Table 4: The number of unique bidding logic scripts we found per URL per interest group owner (note that it is likely that some are serving even more unique variations of their scripts which our crawler did not get by chance)

Seller	Decision logic script URL	Unique scripts
doubleclick.net	`securepubads.g.doubleclick.net/td/sjs`	5
seedtag.com	`t.seedtag.com/ps/seller/decision-logic.js`	1

Table 5: The number of unique decision logic scripts we found per URL per ad auction seller

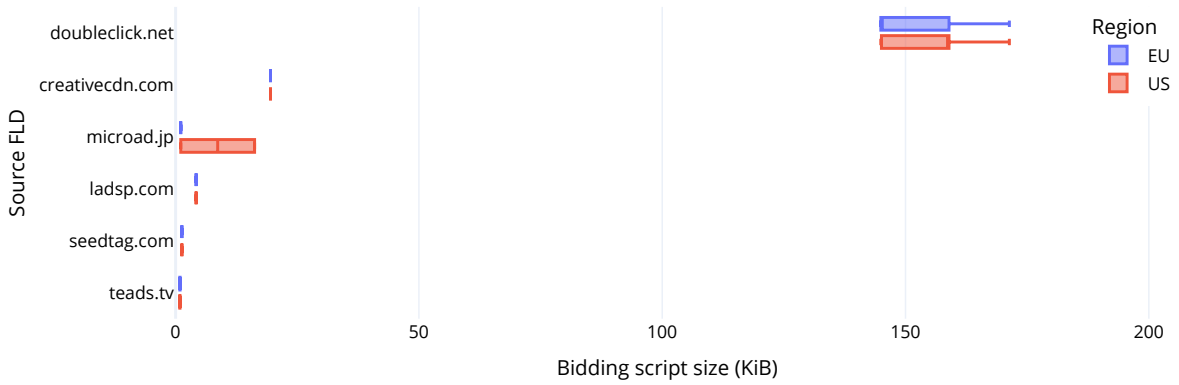

Figure 12: The sizes of the bidding logic scripts per source

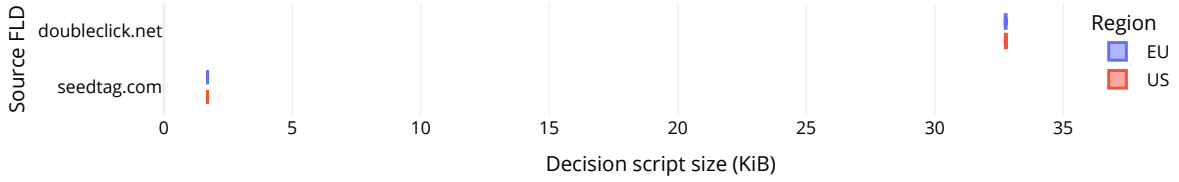

Figure 13: The sizes of the decision logic scripts per source

5 Discussion

In this chapter, we discuss some privacy considerations relevant to the Protected Audience API. In particular, we discuss two main points of concern: external connections and the user's consent. Furthermore, we also discuss why we think Google is now focusing more on privacy with the Privacy Sandbox APIs. Lastly, we discuss some limitations to our research and some possible future research opportunities.

5.1 External connections

While the main appeal of the Protected Audience API is the fact that it does ad auctions on the user's device, there is still a need to communicate with external services:

- An ad might want to check the remaining campaign budget before generating a bid
- An owner might want to update their interest group, e.g. to add or remove ads
- When an ad wins an ad auction, the seller and the buyer need to be informed such that the buyer can pay the seller for the ad space
- The winning ad needs to be loaded from somewhere to be displayed to the user

In general, keeping the raw data on the user's device does not guarantee privacy, as the usage of inferred data may still violate the user's privacy [35]. Some external connections might only be used to load data, such as the interest group's update URL used to update e.g. the ads associated with the interest group. However, this could still expose a privacy risk of your IP address being linked to a specific interest group. So how is the user's privacy ensured despite all these connections to external services?

5.1.1 Key/Value services

During an ad auction, the buyer and the seller can use real-time data from key/value services. The buyer can use this data in their bidding logic to generate a bid, and the seller can use their data in the decision logic to score ads. This data is used for example by buyers to check the remaining campaign budget. To use the data from a key/value service, a `trustedBiddingSignalsUrl` or a `trustedScoringSignalsUrl` must be provided as part of the interest group or ad auction. The browser will automatically load data from these URLs and pass the data on to the `generateBid` and `scoreAd` functions.

During our crawl, we found that all calls to `joinAdInterestGroup` included a `trustedBiddingSignalsUrl`, except those from `ladsp.com`. Furthermore, all calls to `runAdAuction` included a `trustedScoringSignalsUrl`, except those from `seedtag.com`. The owners and sellers always seem to use the same bidding/scoring signals URL for all of their interest groups and ad auctions. For example, DoubleClick always uses `https://td.doubleclick.net/td/bts` for their trusted bidding signals.

These key/value services are hosted by the advertising companies that own the interest group or run the ad auction. Currently, during the initial roll-out of the Protected Audience API, these services can be provided by any server running any software from these advertising companies, as long as they implement the expected API interface. This is a privacy concern, as this could allow these companies to collect data about e.g. when an IP address participates in an ad auction with a certain interest group.

To address this concern, Google plans on requiring these services to run in a *Trusted Execution Environment* (TEE). A TEE is an environment that provides data integrity, data confidentiality, and code integrity [18, 34]. This makes it possible to verify that a service is running known software, and it guarantees that data does not leave the service in unintended ways. To host a key/value service for the Protected Audience API, the server must then run the open-source key/value service [49] in a TEE, which can then be verified by the browser.

This open-source key/value service allows advertising companies to update their data in real time, but it prevents them from collecting any user data. While advertisers might be able to analyze network traffic to see which IP addresses interact with the key/value service, the data sent to the TEE (e.g. metadata of interest groups that the user is in) will be encrypted. The data confidentiality property of TEEs ensures that the decrypted data cannot leave the TEE, which means that this data can only be used to look up data within the TEE.

These TEEs seem great for protecting the user's privacy while looking up real-time external data, such as the remaining budget of an advertising campaign. However, the key/value services are not required to run in a TEE until at least Q3 2025 [60]. Until then, advertisers can track the data sent to key/value services by using their own arbitrary implementations.

5.1.2 k-anonymity services

While TEEs can guarantee the privacy of data sent to key/value services, they do not resolve all privacy concerns. After an ad wins an ad auction, the winning ad needs to be loaded in order to be displayed to the user. This could potentially make a user identifiable to the ad company. E.g. by creating a unique interest group for every user, the ad company could potentially track users across different sites using ad auctions.

To prevent interest groups from being used to track users, the Protected Audience API enforces k-anonymity. Data is said to satisfy k-anonymity if any attempt to identify a user with that data leads to at least k different users [66]. For the Protected Audience API, an ad satisfies k-anonymity if the combination of the interest group owner, bidding logic script URL, render URL, and ad size is shared among at least k different users during the last 30 days [50]. If an ad wins that does not satisfy this k-anonymity, then it will not be rendered and instead, the next highest bid from the ad auction will be picked. This makes sure that when an ad is rendered, at least k other users have the same ad in their interest groups.

The requirements for k-anonymity will slowly ramp up in the coming years. Currently, the ad size is not a part of the k-anonymity check, but it will be in the future. Right now, they use $k = 10$, which means that 10 different users must encounter the same ad based on the previously mentioned properties before the ad is shown. This threshold is planned to increase to $k = 50$ in the future [50].

To check if an ad satisfies k-anonymity, Google hosts a k-anonymity service [79]. A user's browser will regularly tell this service which ads the user has in their interest groups. It does this by sending the hashes of the ad data together with an identifier for the browser to the k-anonymity service. Before the browser shows an ad, it will first query this service with the hash of the ad data. The service will then tell the browser if this ad is k-anonymous by checking if it has recently seen that hash at least k times from different users.

Because the data is hashed, the k-anonymity service does not have access to the ad data. This means that the k-anonymity service cannot see which exact ads a user might be interested in. Additionally, the identifiers that the browsers use will be of low entropy [79], which means they are not unique but are shared by many random users. There are also plans to use Oblivious HTTP [75] to hide the IP addresses where the requests are coming from.

The endpoint used to register a user's ads will be further protected using the Private State Tokens API [54]. This aims to prevent bots from artificially reporting ads to the k-anonymity service, which could otherwise make an ad reach the k-anonymity threshold quicker. Since the creation of Private State Tokens requires the user to be logged in to their Google account, users who are not logged in will not contribute to the k-anonymity threshold [79]. They can, however, still query the k-anonymity service to check if an ad is k-anonymous or not.

Since the k-anonymity service is owned and operated by Google, it will only be available in Google Chrome [79]. If other browsers also want to support the Protected Audience API, they would have to host their own k-anonymity service to enforce k-anonymity. This could potentially be a bit problematic, as ads on less popular browsers with fewer users will take longer to reach the same k-anonymity threshold.

Initially, there were also plans to enforce k-anonymity for interest group updates using the daily update URL. This would prevent the ads retrieved from the update to be specifically targeted to the user. However, calls to `joinAdInterestGroup` can contain ads that are targeted to the users based on first-party user data. Because of this, advertisers would be incentivized to create many interest groups for specific groups of users. This would also mean that users are added to many interest groups, which would increase latency during ad auctions. So in the end, they decided not to enforce k-anonymity for the daily update URLs, and thus to allow the update URLs to contain user IDs. However, the impact on privacy is minimal, as the interest group update requests only contain data from the original site where the user was added to the interest group [81].

5.1.3 Auction reporting

While the k-anonymity requirements prevent ads from being used to re-identify an individual user, the results of the ad auction still need to be reported to the advertising companies. Currently, auction results can be reported using the `sendReportTo` method [55]. This method allows the buyer and the seller to specify a URL of their choosing to which a request will be sent after they win an ad auction. The URL can include various data as URL parameters, like the bid itself, the interest group owner, or various auction signals. An example of this can be seen in figure 14, which shows how `seedtag.com` was using this method in a bidding script.

With online advertising, companies often only pay for clicks or conversions [84]. This is better for the advertisers, as they only pay for an ad space when a user is truly interested in their advertisements. Additionally, it also incentivizes the DSP to select the most relevant advertisement for the user during the ad auction. However, to facilitate this payment system, various user events, like clicks or conversions, also need to be reported to the advertising companies.

```
const event = 'result';
const qp = {
    event,
    bid: browserSignals.bid,
    interestGroupOwner: browserSignals.interestGroupOwner,
    currency: browserSignals.bidCurrency || 'USD',
    desirability: browserSignals.desirability,
    token: auctionConfig?.sellerSignals?.publisherToken || 'Other',
    productShortCode: auctionConfig?.sellerSignals?.productShortCode || 'Other',
};
const qpString = `event=${event}&bid=${qp.bid}&interestGroupOwner=${qp.interestGroupOwner}&currency=${qp.currency}&
                  desirability=${qp.desirability}&token=${qp.token}&productShortCode=${qp.productShortCode}`;
sendReportTo(`https://s.seedtag.com/e/pa?${qpString}`);
```

Figure 14: A fragment from the `reportWin` function found in a bidding scripts from `seedtag.com`, showing how `sendReportTo` is used to report various auction data after winning the ad auction

Normally, advertisements would send these user events back to the advertiser themselves, using identifiers from the main page to associate the events with the ad auction. However, the Privacy Sandbox plans to require ads to be rendered in Fenced Frames [48]. Fenced Frames are similar to Inline Frames (`<iframe>` in HTML), which can embed other websites on a page, but with restricted communication between the main page and the embedded page. Because of these restrictions, events from within a Fenced Frame cannot be associated with the ad auction that took place on the main page. It is not possible to share e.g. a user ID without any communication with the main page, while also satisfying the k-anonymity requirements regarding the render URLs, as discussed in section 5.1.2.

Instead, the Ads Reporting API can be used to report events from within Fenced Frames in such a way that they can be associated with the results of the ad auction. The Ads Reporting API allows advertisers to register an ad beacon, to which events are sent from within Fenced Frames [55]. The `registerAdBeacon` method is used to set the URLs to which various events, such as clicks, impressions, and errors, will be sent by the browser. Figure 15 shows how this method was used by `teads.tv` in one of the bidding scripts we collected during our crawl.

```
registerAdBeacon({
    click:       `${s.interestGroupOwner}/track?action=click&auctid=${e.auctionId}`,
    impression:  `${s.interestGroupOwner}/track?action=impression&auctid=${e.auctionId}`,
    earlyClick:  `${s.interestGroupOwner}/track?action=early-click&auctid=${e.auctionId}`,
    error:       `${s.interestGroupOwner}/track?action=error-vast&auctid=${e.auctionId}`,
    "visible-1": `${s.interestGroupOwner}/track?action=visible-1&auctid=${e.auctionId}`,
    "visible-2": `${s.interestGroupOwner}/track?action=visible-2&auctid=${e.auctionId}`,
    "visible-3": `${s.interestGroupOwner}/track?action=visible-3&auctid=${e.auctionId}`,
    "visible-4": `${s.interestGroupOwner}/track?action=visible-4&auctid=${e.auctionId}`,
    "visible-5": `${s.interestGroupOwner}/track?action=visible-5&auctid=${e.auctionId}`
})
```

Figure 15: A fragment from the `reportWin` function found in a bidding script from `teads.tv`, showing how `registerAdBeacon` is used to configure URLs to which events from the advertisement will be reported

The Ads Reporting API and the `sendReportTo` method are both temporary solutions to make the transition to the Protected Audience API easier for advertisers. These functions are available until at least 2026, and Fenced Frames will not be required until at least 2026 [60]. In the future, advertisers will need to switch to more privacy-friendly APIs to keep track of auction results and user events from advertisements. The Private Aggregation API will be used in the future to generate reports of auction results, where these reports will combine data from many auctions [53]. The Attribution Reporting API will be used to track conversions and other events, by allowing advertisers to request event reports from the user's browser [44]. However, these APIs do not need to be used until at least 2026, as until then, simpler (and less privacy-friendly) alternatives are available for advertisers.

5.2 User consent

During our crawl, we used Tracker Radar Collector's CMP collector to try to automatically give consent when a website asks if it can store cookies. This collector successfully gave consent on 2,717 websites during our 10k-site crawl. But does giving consent actually influence whether websites use the Protected Audience API?

In section 4.2 we saw that there were many more calls to `joinAdInterestGroup` in the US compared to our EU crawl. This could be due to privacy regulations such as the GDPR, which limit what companies are allowed to do with personal data of users in the EU. The GDPR requires consent for tracking cookies used for Real-Time Bidding, as they can be used to identify the user [78]. According to the GDPR, any processing of personal data requires consent, unless there are other grounds for the processing [25].

It is unclear if interest groups would be affected by the GDPR in the same way. In theory, interest groups could contain some amount of personal information, e.g. as part of the `userBiddingSignals` provided when calling `joinAdInterestGroup`. However, the Privacy Sandbox aims to prevent ads from tracking individual users, using e.g. Fenced Frames, as discussed in section 5.1. This would mean that no personal data should leave the user's device when all the privacy requirements of the Privacy Sandbox are enabled in the future.

Other than the GDPR, there is also the ePrivacy Directive which applies in the EU[3]. Article 5(3) of the ePrivacy Directive prevents websites from storing or gaining access to data (including cookies) on the user's device without consent, unless strictly necessary [24]. As this article is purposely specified without mentioning any specific technology, it will most likely also apply to interest groups, which are stored on the user's device.

Google seems to agree with this sentiment, as they have included mentions of their new Privacy Sandbox APIs in their consent policies in the EU [52]. We did a small manual experiment with the ten sites that had the most calls to `joinAdInterestGroup` in our EU crawl, and we noticed most sites will not add you to an interest group unless you provide consent. Only one of the ten sites we tested (`banggood.com`, a Chinese web shop selling goods worldwide) added you to interest groups without providing consent.

[3]The ePrivacy Directive is a *directive*, not a regulation like the GDPR, so implementations of this directive might differ from country to country within the EU. The EU is working on an ePrivacy Regulation, which aims to modernize the ePrivacy Directive and make it directly applicable like the GDPR [23].

When a user's version of Chrome is updated to support the new Privacy Sandbox APIs, or when they download a new version of Chrome that supports them, the user is informed about the new APIs by a pop-up. This pop-up, as seen in figure 16, asks users in the EU if they want to enable the Topics API or not. However, the Protected Audience API and the Attribution Reporting API are on by default, although there is a button to open the settings where they can be turned off. According to Google, they did this because the Topics API is a completely new system, while the Protected Audience API and the Attribution Reporting API are just more private versions of systems that were already being used previously [52].

After you are added to an interest group on a website, you can opt out of interest groups from that specific site by blocking the site in the Chrome settings. However, there is no easy way to see any details about the interest groups that you have been added to. You cannot see which third parties manage these interest groups. You also do not know if the interest groups are about a specific product, or if they are more general interest groups with many ads for different products. We think more transparency in this regard would be appreciated, for example by having an internal page with more details, similar to the Topics API[4].

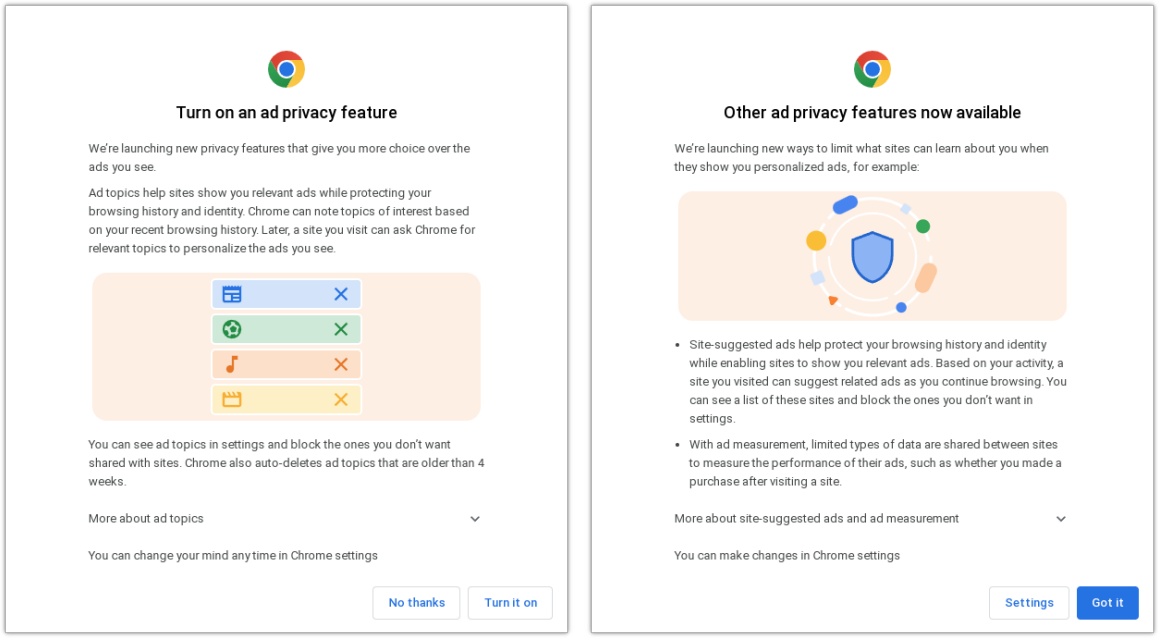

Figure 16: Pop-up in Google Chrome, first asking if the user wants to enable the Topics API or not (on the left), and then explaining that the Protected Audience API and ad measurement APIs are enabled and can be turned off in the Chrome settings (on the right)

[4]For the Topics API, `chrome://topics-internals/` shows which topics the browser thinks the user might be interested in, and it shows additional information about how it has inferred these topics based on your browsing behavior.

5.3 Google's potential motivation

So far, we have discussed how Google is facilitating targeted advertising in a more privacy-friendly manner using the Protected Audience API. As we saw during our crawl, Google is the most prevalent user of their own Privacy Sandbox APIs. Advertising is the biggest source of income for Google, as 76.6% of the revenue of the parent company Alphabet consists of advertising revenue (as of Q1 2024) [4]. However, disabling third-party cookies is expected to decrease the effectiveness of targeted advertising [3, 26]. So why is Google developing the Privacy Sandbox, and why are they disabling third-party cookies in the first place?

We think that one of the main reasons why they are now focusing on privacy could be because both users and legislators have become more privacy-aware over the last few years. As a result of the GDPR and the ePrivacy Directive, users get confronted with cookie popups all over the web. As people are given the choice, some users will deny consent. Other more privacy-aware users might go further by manually disabling third-party cookies in their browsers. Some users might even install ad or tracker blockers, either for privacy reasons or because they simply do not want to deal with all the popups and banners.

Users opting out of targeted advertising or using ad blockers greatly impacts advertising revenue [31, 63]. While advertisers can try to detect and block users using ad blockers, this will likely have a negative effect on user engagement [87]. Advertisers may also try to circumvent ad block rules [71], but this just results in a never-ending battle between the advertisers and the ad blocker developers. When Google announced stricter requirements for browser extensions (known as Manifest V3 [14]), this caused quite some controversy as this limited the capabilities of ad blockers [70, 6]. It is clear that a good number of internet users care a lot about ad blockers, and it is difficult to get them to view ads.

With the Privacy Sandbox, Google seems to be taking a different approach. By focusing on the user's privacy, we speculate they might hope to convince both users and legislators that targeted advertisements do not have to be a bad thing. If there are fewer reasons for users to be concerned about advertising companies tracking their behavior online, then users have fewer reasons to install an ad blocker or deny consent. Additionally, Google has mentioned that they hope consent regulations might be loosened for privacy-friendly technologies in the future [52]. While we can only speculate about Google's true motivations, the Privacy Sandbox might be a way for Google to ensure that they can keep benefitting from targeted advertising in the foreseeable future.

5.4 Limitations

As our crawler is based on Tracker Radar Collector [20] with an API call collector based on a script by Senol and Acar [69], many similar limitations apply to our work. For example, while crawling, the crawler might get blocked by a website's anti-bot protection. While Tracker Radar Collector has some built-in countermeasures against anti-bot protection, these might not always work perfectly. Our crawler could also be limited because it is running on a Digital Ocean server. In figure 5 we saw a higher percentage of failed crawls in the US compared to the EU. This could be due to our US IP being blocked because of prior suspicious activity from other Digital Ocean users.

Overall, our results are based on a single visit to each website per crawl location, with only a limited amount of interaction on the site. Our crawler does scroll the page and navigate to one subpage, but this does not guarantee that we catch all API calls that might get triggered on a website. Additionally, ads tend to be probabilistic, and might not always show up in the same amount when visiting a page. Doing multiple visits to a website and crawling more subpages per website could lead to more complete results, but it would also drastically increase the duration of the crawl.

5.5 Future work

We did our crawl at the end of 2023. At that time, Chrome had not yet started blocking third-party cookies, and we only got a limited amount of results for the Protected Audience API. It would be interesting to repeat our crawling experiment a year or two from now to see how things have changed, especially since Chrome should at that point be blocking third-party cookies for everyone by default. It will also be interesting to see how the stricter privacy requirements that have been promised for the Protected Audience API will turn out.

Most of our results for the Protected Audience API calls came from DoubleClick, unlike the results from the Topics API, which came from many more different sources. Unlike the Topics API, the Protected Audience API requires additional infrastructure, such as key/value services, to be used effectively. Because of this, we do not expect to see a lot of sites calling this API themselves. Instead, most calls will likely always go through a larger DSP like Google's DoubleClick. A larger crawl of more than 10,000 websites could be useful to get a more complete picture of how the APIs are used on less popular sites to confirm this hypothesis. However, such a crawl would require more powerful hardware (e.g. more bandwidth and more CPU cores) to finish the crawl in a decent amount of time.

Our analysis of the auction logic scripts only touched on some surface-level comparisons and the inspection of some basic test scripts that we came across during our crawl. Future work could look further into the logic within the larger scripts, such as those from DoubleClick. However, it can be difficult to understand what is going on, as these scripts tend to be minified, and it might be unclear what kind of data goes into them (e.g. data requested from key/value services). Additionally, they can also make use of WebAssembly binaries, which would require even more reverse engineering to fully comprehend the auction logic.

We investigated the Protected Audience API for the web, but Google's Privacy Sandbox also has similar APIs for Android apps. The Android version of the Protected Audience API also includes various phone-specific features, such as filtering ads based on the user's installed apps [57]. These Android APIs could also be interesting to investigate further, as there might be more privacy concerns regarding these mobile APIs.

In this paper, we did not go into depth on some of the APIs that the Protected Audience API can interact with. More in-depth analysis could for example be done regarding the reporting APIs mentioned in section 5.1.3. Some features of the Privacy Sandbox, like the Private Aggregation API, are still in active development and have not yet been released to the public. Other Privacy Sandbox APIs could also be updated in the future with new functionalities, e.g. to appease the advertising industry as a result of the ongoing investigation from the CMA [7]. It would be good to pay close attention to how Google further develops the Privacy Sandbox in the future.

6 Conclusion

We crawled 10,000 sites to get an early look at the usage of Google's new Protected Audience API. We did our crawl from both the US and the EU to get an idea of how the region might affect the usage of the API on websites. In total, our crawler collected 6,594 calls to the Protected Audience API from seven different advertising companies on 809 different sites. Most of these calls came from DoubleClick, Google's own online advertising platform.

Our crawler also collected calls to the Topics API, another API from Google's Privacy Sandbox. We collected about three times more calls to the Topics API (20,317 in total on 2,572 different sites), and these calls came from many more different parties (65 to be exact). This difference is likely because Topics API has more general use cases, unlike the Protected Audience API, which is specifically for remarketing. Furthermore, the Topics API is easier to use, as it does not require e.g. key/value services to be set up to use it effectively.

We saw some slightly unusual usage of the Protected Audience API. Retargetly seemingly only wanted users to leave their interest groups, as we did not get any calls from them to join their interest groups, while they did call to leave. We also saw an interest group from RTB House with a duration of 30,000 days, which is a thousand times more than the maximum allowed 30 days. Some companies, like DoubleClick, were using large complicated minified auction logic scripts for bidding and scoring ads. Others were using very simple scripts, indicating that they were likely only experimenting and testing out the Protected Audience API at the time of the crawl.

The main appeal of the Protected Audience API is that it does ad auctions on the user's own device, which means that advertisers no longer need to track the user to target their advertisements. By using Trusted Execution Environments, k-anonymity services, and aggregated event reporting APIs they protect the user's privacy when data does have to leave the user's device. However, to ease the transition to these new APIs for advertising companies, many privacy requirements are currently relatively weak. Additionally, the transparency of the Protected Audience API is something that could be improved, as users have access to very little information about the interest groups they are added to.

In the EU, the GDPR and the ePrivacy Directive require advertisers to ask for consent before using tracking cookies. These same rules also apply to new technologies, including the new Privacy Sandbox APIs. This seems to be reflected in our data, as we found more than three times as many websites adding users to interest groups in the US compared to the EU, despite having a higher crawl success rate in the EU. By focusing more on the user's privacy, we speculate Google might be trying to regain the trust of users and regulators. In the end, this might reduce the loss in revenue from users blocking ads or opting out of tracking for Google's own targeted advertising services.

In general, Google's Privacy Sandbox seems to be a step in the right direction when it comes to the user's privacy. While right now the privacy requirements for the Protected Audience API are still relatively weak, these restrictions will get stricter in the future to better protect the user's privacy. The API is already much better for the user's privacy compared to unrestricted third-party tracking cookies, which do not provide any privacy guarantees for the user at all. However, as these Privacy Sandbox APIs are used for targeted advertising, they will always involve some amount of user tracking, and it is up to the user to decide whether they find that acceptable or not.

Bibliography

[1] Karan Aggarwal, Pranjul Yadav, and S. Sathiya Keerthi. "Domain Adaptation in Display Advertising: An Application for Partner Cold-Start". In: *Proceedings of the 13th ACM Conference on Recommender Systems*. RecSys '19. Copenhagen, Denmark: Association for Computing Machinery, 2019, pp. 178–186. ISBN: 9781450362436. DOI: 10.1145/3298689.3347004.

[2] Hidayet Aksu et al. *Summary Reports Optimization in the Privacy Sandbox Attribution Reporting API*. Google, 2023. arXiv: 2311.13586 [cs.CR].

[3] Miguel Alcobendas et al. *The Impact of Privacy Protection on Online Advertising Markets*. Yahoo Research, Oct. 2023. DOI: 10.2139/ssrn.3782889.

[4] Alphabet Inc. *Alphabet Announces First Quarter 2024 Results*. Apr. 25, 2024. URL: https://www.abc.xyz/assets/91/b3/3f9213d14ce3ae27e1038e01a0e0/2024q1-alphabet-earnings-release-pdf.pdf (visited on 04/29/2024).

[5] Mário S. Alvim et al. *A Quantitative Information Flow Analysis of the Topics API*. 2023. arXiv: 2309.14746 [cs.CR].

[6] Ron Amadeo. "Chrome's "Manifest V3" plan to limit ad-blocking extensions is delayed". In: *Ars Technica* (Dec. 13, 2022). URL: https://arstechnica.com/gadgets/2022/12/chrome-delays-plan-to-limit-ad-blockers-new-timeline-coming-in-march/ (visited on 04/29/2024).

[7] Competition & Markets Authority. *CMA Q1 2024 Update report on implementation of the Privacy Sandbox commitment*. Apr. 2024. URL: https://assets.publishing.service.gov.uk/media/662baa3efee48e2ee6b81eb1/1._CMA_Q1_2024_update_report_on_Google_Privacy_Sandbox_commitments.pdf (visited on 04/29/2024).

[8] Alex Berke and Dan Calacci. "Privacy Limitations of Interest-Based Advertising on The Web: A Post-Mortem Empirical Analysis of Google's FLoC". In: *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. CCS '22. Los Angeles, CA, USA: Association for Computing Machinery, 2022, pp. 337–349. ISBN: 9781450394505. DOI: 10.1145/3548606.3560626.

[9] Yohan Beugin and Patrick McDaniel. *A Public and Reproducible Assessment of the Topics API on Real Data*. 2024. arXiv: 2403.19577 [cs.CR].

[10] Yohan Beugin and Patrick McDaniel. *Interest-disclosing Mechanisms for Advertising are Privacy-Exposing (not Preserving)*. 2023. arXiv: 2306.03825 [cs.CR].

[11] "Building a more private web: A path towards making third party cookies obsolete". In: *Chromium Blog* (Jan. 14, 2020). URL: https://blog.chromium.org/2020/01/building-more-private-web-path-towards.html (visited on 06/14/2024).

[12] Dave Camp. "Firefox Now Available with Enhanced Tracking Protection by Default Plus Updates to Facebook Container, Firefox Monitor and Lockwise". In: *Mozilla Blog* (June 4, 2019). URL: https://blog.mozilla.org/en/products/firefox/firefox-now-available-with-enhanced-tracking-protection-by-default/ (visited on 06/21/2024).

[13] Anthony Chavez. "Privacy Sandbox for the Web reaches general availability". In: *Privacy Sandbox* (Sept. 7, 2023). URL: https://privacysandbox.com/news/privacy-sandbox-for-the-web-reaches-general-availability/ (visited on 06/14/2024).

[14] Chrome for Developers. *Manifest V3*. URL: https://developer.chrome.com/docs/extensions/develop/migrate/what-is-mv3 (visited on 04/29/2024).

[15] Chrome Platform Status. *Usage metrics for the `joinAdInterestGroup` method*. URL: https://chromestatus.com/metrics/feature/timeline/popularity/3855 (visited on 01/26/2024).

[16] Chrome Platform Status. *Usage metrics for the `runAdAuction` method*. URL: https://chromestatus.com/metrics/feature/timeline/popularity/3857 (visited on 01/26/2024).

[17] Chrome Platform Status. *Usage metrics for the `updateAdInterestGroups` method*. URL: https://chromestatus.com/metrics/feature/timeline/popularity/3904 (visited on 01/26/2024).

[18] Confidential Computing Consortium. *Confidential Computing: Hardware-Based Trusted Execution for Applications and Data (V1.3)*. Nov. 2022. URL: https://confidentialcomputing.io/wp-content/uploads/sites/10/2023/03/CCC_outreach_whitepaper_updated_November_2022.pdf.

[19] Dylan Cooper et al. "Privacy considerations for online advertising: A stakeholder's perspective to programmatic advertising". In: *Journal of Consumer Marketing* ahead-of-print (Dec. 2021). DOI: 10.1108/JCM-04-2021-4577.

[20] DuckDuckGo. *Tracker Radar Collector*. URL: https://github.com/duckduckgo/tracker-radar-collector (visited on 09/18/2023).

[21] "DuckDuckGo Tracker Radar Exposes Hidden Tracking". In: *Spread Privacy: DuckDuckGo News* (Mar. 5, 2020). URL: https://spreadprivacy.com/duckduckgo-tracker-radar/ (visited on 06/21/2024).

[22] David Eliot and David Murakami Wood. "Culling the FLoC: Market forces, regulatory regimes and Google's (mis)steps on the path away from targeted advertising". In: *Information Polity* 27 (2022). 2, pp. 259–274. ISSN: 1875-8754. DOI: 10.3233/IP-211535.

[23] European Commision. *Proposal for an ePrivacy Regulation*. URL: https://digital-strategy.ec.europa.eu/en/policies/eprivacy-regulation (visited on 05/20/2024).

[24] European Parliament and Council of the European Union. *Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications)*. 2002. URL: https://eur-lex.europa.eu/eli/dir/2002/58/2009-12-19.

[25] 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)*. 2016. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679.

[26] IAB Tech Lab's Privacy Sandbox Task Force. *Privacy Sandbox: Fit Gap Analysis for Digital Advertising*. Feb. 6, 2024. URL: https://iabtechlab.com/privacysandbox/ (visited on 06/14/2024).

[27] Przemysław Iwańczak and Mateusz Rumiński. "The Future of Frequency Capping in Privacy-Centric Digital Advertising". In: *SSRN Electronic Journal* (Jan. 2022). DOI: 10.2139/ssrn.3985974.

[28] Kinshuk Jerath and Klaus M. Miller. *Using the Dual-Privacy Framework to Understand Consumers' Perceived Privacy Violations Under Different Firm Practices in Online Advertising*. 2024. arXiv: 2403.03612 [econ.GN].

[29] Nikhil Jha et al. *On the Robustness of Topics API to a Re-Identification Attack*. 2023. arXiv: 2306.05094 [cs.CY].

[30] Garrett Johnson, Julian Runge, and Eric Seufert. "Privacy-Centric Digital Advertising: Implications for Research". In: *Customer Needs and Solutions* 9.1 (June 2022), pp. 49–54. ISSN: 2196-2928. DOI: 10.1007/s40547-022-00125-4.

[31] Garrett A. Johnson, Scott K. Shriver, and Shaoyin Du. "Consumer Privacy Choice in Online Advertising: Who Opts Out and at What Cost to Industry?" In: *Marketing Science* 39.1 (2020), pp. 33–51. DOI: 10.1287/mksc.2019.1198.

[32] Florian Lachner, Minzhe Cheng, and Theodore Olsauskas-Warren. "User Attitudes Towards Controls for Ad Interests Estimated On-device by the Browser". In: *Proceedings 2023 Symposium on Usable Security*. USEC 2023. Google. Feb. 2023. DOI: 10.14722/usec.2023.239417.

[33] Victor Le Pochat et al. "Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation". In: *Proceedings of the 26th Annual Network and Distributed System Security Symposium*. NDSS 2019. Feb. 2019. DOI: 10.14722/ndss.2019.23386.

[34] Pieter Maene et al. "Hardware-Based Trusted Computing Architectures for Isolation and Attestation". In: *IEEE Transactions on Computers* 67.3 (2018), pp. 361–374. DOI: 10.1109/TC.2017.2647955.

[35] Kirsten Martin, Helen F. Nissenbaum, and Vitaly Shmatikov. *No Cookies For You!: Evaluating The Promises Of Big Tech's 'Privacy-Enhancing' Techniques*. Dec. 2023. DOI: 10.2139/ssrn.4655228.

[36] Lee McGuigan et al. "Private attributes: The meanings and mechanisms of "privacy-preserving" adtech". In: *New Media & Society* (Nov. 2023). DOI: 10.1177/14614448231213267.

[37] Microsoft. *PARAKEET*. URL: https://github.com/microsoft/PARAKEET (visited on 06/23/2024).

[38] S. Muthukrishnan. "AdX: A Model for Ad Exchanges". In: *SIGecom Exch.* 8.2 (Dec. 2009). DOI: 10.1145/1980522.1980531.

[39] Mark Nottingham. *Playing Fair in the Privacy Sandbox: Competition, Privacy and Interoperability Standards*. 2021. DOI: 10.2139/ssrn.3891335.

[40] "OK Google, don't delay real browser privacy until 2022". In: *Brave Blog* (Feb. 6, 2020). URL: https://brave.com/blog/ok-google/ (visited on 06/14/2024).

[41] Lukasz Olejnik. "On the governance of privacy-preserving systems for the web: should Privacy Sandbox be governed?" In: *Handbook on the Politics and Governance of Big Data and Artificial Intelligence*. Ed. by Andrej Zwitter and Oskar J. Gstrein. Cheltenham, UK: Edward Elgar Publishing, 2023. Chap. 10, pp. 279–314. DOI: 10.4337/9781800887374.00022.

[42] Weitong Ou et al. "A Survey on Bid Optimization in Real-Time Bidding Display Advertising". In: *ACM Trans. Knowl. Discov. Data* (Oct. 2023). Just Accepted. ISSN: 1556-4681. DOI: 10.1145/3628603.

[43] Emmanouil 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*. ACM, Apr. 2021. DOI: 10.1145/3442381.3450056.

[44] Privacy Sandbox. *Attribution Reporting API documentation*. Google. URL: https://developers.google.com/privacy-sandbox/relevance/attribution-reporting (visited on 05/15/2024).

[45] Privacy Sandbox. *Cookies Having Independent Partitioned State (CHIPS) documentation*. Google. URL: https://developer.chrome.com/docs/privacy-sandbox/chips/ (visited on 11/30/2023).

[46] Privacy Sandbox. *Define audience data*. Google. URL: https://developers.google.com/privacy-sandbox/relevance/define-audience-data (visited on 04/30/2024).

[47] Privacy Sandbox. *Federated Credential Management API documentation*. Google. URL: https://developer.chrome.com/docs/privacy-sandbox/fedcm/ (visited on 11/30/2023).

[48] Privacy Sandbox. *Fenced frames overview*. Google. URL: https://developers.google.com/privacy-sandbox/relevance/fenced-frame (visited on 05/14/2024).

[49] Privacy Sandbox. *FLEDGE Key/Value service*. Google. URL: https://github.com/privacysandbox/protected-auction-key-value-service (visited on 04/25/2024).

[50] Privacy Sandbox. *K-anonymity*. Google. URL: https://developers.google.com/privacy-sandbox/relevance/protected-audience-api/k-anonymity (visited on 04/30/2024).

[51] Privacy Sandbox. *Prepare for the third-party cookie phaseout*. Google. URL: https://developers.google.com/privacy-sandbox/3pcd/prepare/prepare-for-phaseout (visited on 04/29/2024).

[52] Privacy Sandbox. *Privacy-related compliance FAQs*. URL: https://developers.google.com/privacy-sandbox/overview/privacy-compliance-faqs (visited on 05/20/2024).

[53] Privacy Sandbox. *Private Aggregation API documentation*. Google. URL: https://developers.google.com/privacy-sandbox/relevance/private-aggregation (visited on 05/15/2024).

[54] Privacy Sandbox. *Private State Tokens API documentation*. Google. URL: https://developer.chrome.com/docs/privacy-sandbox/private-state-tokens/ (visited on 11/30/2023).

[55] Privacy Sandbox. *Protected Audience API auction reporting*. Google. URL: https://developers.google.com/privacy-sandbox/relevance/protected-audience-api/reporting (visited on 05/14/2024).

[56] Privacy Sandbox. *Protected Audience API documentation*. Google. URL: https://developers.google.com/privacy-sandbox/relevance/protected-audience-api (visited on 05/21/2024).

[57] Privacy Sandbox. *Protected Audience API for Android documentation*. Google. URL: https://developers.google.com/privacy-sandbox/relevance/protected-audience/android (visited on 05/14/2024).

[58] Privacy Sandbox. *Related Website Sets documentation*. Google. URL: https://developer.chrome.com/docs/privacy-sandbox/related-website-sets/ (visited on 11/30/2023).

[59] Privacy Sandbox. *Shared Storage API documentation*. Google. URL: https://developer.chrome.com/docs/privacy-sandbox/shared-storage/ (visited on 11/30/2023).

[60] Privacy Sandbox. *Status of pending Protected Audience API capabilities*. Google. URL: https://developers.google.com/privacy-sandbox/relevance/protected-audience-api/feature-status (visited on 05/14/2024).

[61] Privacy Sandbox. *Topics API documentation*. Google. URL: https://developers.google.com/privacy-sandbox/relevance/topics/developer-guide (visited on 01/04/2024).

[62] Privacy Sandbox. *What is the Privacy Sandbox?* Google. URL: https://developer.chrome.com/docs/privacy-sandbox/overview/ (visited on 09/18/2023).

[63] Enric Pujol, Oliver Hohlfeld, and Anja Feldmann. "Annoyed Users: Ads and Ad-Block Usage in the Wild". In: *Proceedings of the 2015 Internet Measurement Conference*. IMC '15. Tokyo, Japan: Association for Computing Machinery, 2015, pp. 93–106. ISBN: 9781450338486. DOI: 10.1145/2815675.2815705.

[64] "Remarketing — what it is, how it works, and more". In: *Adobe Experience Cloud Blog* (Aug. 18, 2023). URL: https://business.adobe.com/blog/basics/what-is-remarketing (visited on 06/21/2024).

[65] Mateusz Rumiński, Przemysław Iwańczak, and Łukasz Włodarczyk. *Findings from the Early Fledge Experiments*. RTB House, Sept. 2022. DOI: 10.2139/ssrn.4219796.

[66] Pierangela Samarati and Latanya Sweeney. "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression". In: *Proceedings of the IEEE Symposium on Research in Security and Privacy (S&P)*. Oakland, CA, May 1998.

[67] Justin Schuh. *Building a more private web*. Google. Aug. 22, 2019. URL: https://blog.google/products/chrome/building-a-more-private-web/ (visited on 05/27/2024).

[68] *Secure Web Addressability Network (SWAN)*. URL: https://github.com/SWAN-community/swan (visited on 06/23/2024).

[69] Asuman Senol and Gunes Acar. "Unveiling the Impact of User-Agent Reduction and Client Hints: A Measurement Study". In: *Proceedings of the 22nd Workshop on Privacy in the Electronic Society*. WPES '23. Copenhagen, Denmark: Association for Computing Machinery, 2023, pp. 91–106. ISBN: 9798400702358. DOI: 10.1145/3603216.3624965.

[70] Aamir Siddiqui. "Google's Manifest V3 will change how ad blocking Chrome extensions work: Is it to cripple them, or is it for security?" In: *XDA Developers* (June 30, 2019). URL: https://www.xda-developers.com/google-chrome-manifest-v3-ad-blocker-extension-api/ (visited on 04/29/2024).

[71] Peter Snyder, Antoine Vastel, and Ben Livshits. "Who Filters the Filters: Understanding the Growth, Usefulness and Efficiency of Crowdsourced Ad Blocking". In: *Proc. ACM Meas. Anal. Comput. Syst.* 4.2 (June 2020). DOI: 10.1145/3392144.

[72] Martin Thomson. "An Analysis of Apple's Private Click Measurement". In: (June 2022). URL: https://mozilla.github.io/ppa-docs/pcm.pdf.

[73] Martin Thomson. "Privacy Preserving Attribution for Advertising". In: *Mozilla Blog* (Feb. 8, 2022). URL: https://blog.mozilla.org/en/mozilla/privacy-preserving-attribution-for-advertising/ (visited on 06/23/2024).

[74] Martin Thomson and Eric Rescorla. "Comments on SWAN and Unified ID 2.0". In: (Aug. 2021). URL: https://mozilla.github.io/ppa-docs/swan_uid2_report.pdf.

[75] Martin Thomson and Christopher A. Wood. *Oblivious HTTP*. RFC 9458. Jan. 2024. DOI: 10.17487/RFC9458.

[76] *Unified ID 2.0 Overview*. URL: https://unifiedid.com/docs/intro (visited on 06/23/2024).

[77] Blase Ur et al. "Smart, useful, scary, creepy: perceptions of online behavioral advertising". In: *Proceedings of the Eighth Symposium on Usable Privacy and Security*. SOUPS '12. Washington, D.C.: Association for Computing Machinery, 2012. ISBN: 9781450315326. DOI: 10.1145/2335356.2335362.

[78] Michael Veale and Frederik Zuiderveen Borgesius. "Adtech and Real-Time Bidding under European Data Protection Law". In: *German Law Journal* 23.2 (2022), pp. 226–256. DOI: 10.1017/glj.2022.18.

[79] WICG. *Privacy Sandbox k-Anonymity Server*. URL: https://github.com/WICG/turtledove/blob/main/FLEDGE_k_anonymity_server.md (visited on 04/30/2024).

[80] WICG. *TURTLEDOVE*. URL: https://github.com/WICG/turtledove/ (visited on 06/23/2024).

[81] WICG. *Why we need multiple IGs per domain*. URL: https://github.com/WICG/turtledove/issues/361 (visited on 04/30/2024).

[82] John Wilander. "Intelligent Tracking Prevention". In: *WebKit Blog* (June 5, 2017). URL: https://webkit.org/blog/7675/intelligent-tracking-prevention/ (visited on 06/21/2024).

[83] John Wilander. "Introducing Private Click Measurement, PCM". In: *WebKit Blog* (Feb. 1, 2021). URL: https://webkit.org/blog/11529/introducing-private-click-measurement-pcm/ (visited on 06/23/2024).

[84] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. "Real-Time Bidding for Online Advertising: Measurement and Analysis". In: *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*. ADKDD '13. Chicago, Illinois: Association for Computing Machinery, 2013. ISBN: 9781450323239. DOI: 10.1145/2501040.2501980.

[85] Yong Yuan et al. "A survey on real time bidding advertising". In: *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics*. 2014, pp. 418–423. DOI: 10.1109/SOLI.2014.6960761.

[86] Weinan Zhang, Shuai Yuan, and Jun Wang. "Optimal Real-Time Bidding for Display Advertising". In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '14. New York, New York, USA: Association for Computing Machinery, 2014, pp. 1077–1086. ISBN: 9781450329569. DOI: 10.1145/2623330.2623633.

[87] Shuai Zhao et al. "To be Tough or Soft: Measuring the Impact of Counter-Ad-blocking Strategies on User Engagement". In: *Proceedings of The Web Conference 2020*. WWW '20. Taipei, Taiwan: Association for Computing Machinery, 2020, pp. 2690–2696. ISBN: 9781450370233. DOI: 10.1145/3366423.3380025.

A Experiment: crawling multiple subpages

Our crawler crawls subpages using inner links found on the main page. Links are ordered based on how close they are to the center of the screen, as these are most likely to be links to subpages that the website wants users to visit. For example, a web shop is likely to feature products in the center of the screen, while the header and the footer tend to contain links to more "informational" pages, which are less likely to use the Privacy Sandbox APIs.

To determine how many subpages we should crawl for our final crawl, we did a smaller pilot crawl of only 1,000 sites. In this smaller crawl, we crawled two subpages from every website. As seen in figure 17, we collected a good number of calls from the subpages. While it seems that `browsingTopics` was called substantially more on the main page compared to the subpages, the calls to the Protected Audience API were more equally spread out among the (sub-)pages. Unfortunately, this pilot crawl did not collect any calls to `runAdAuction`.

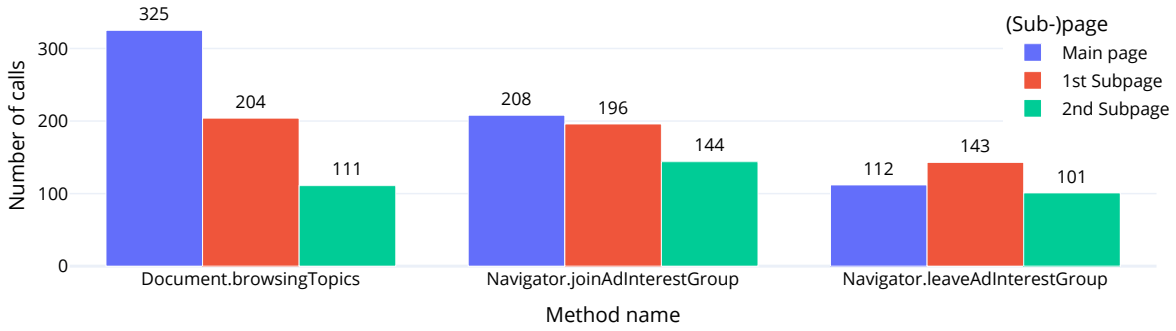

Figure 17: The total number of calls collected per (sub-)page

While it is good to know that the subpages give us plenty of additional calls, there is a trade-off with the time it takes to crawl these subpages. If we crawl fewer subpages, we will be able to crawl more websites in the same amount of time. For this project, we are mostly interested in seeing how different websites and advertisers use the Protected Audience API. This is why we also looked at on how many websites we encountered an API call for the first time on the first or second subpage. Figure 18 shows that for most websites, the first occurrence of an API call came from the main page. However, it also shows that some websites that did not call an API on the main page, did call it on one of their subpages. The second subpage we crawled added very few new results compared to the first subpage. Because of this, we decided to only crawl a single subpage per website for our final crawl.

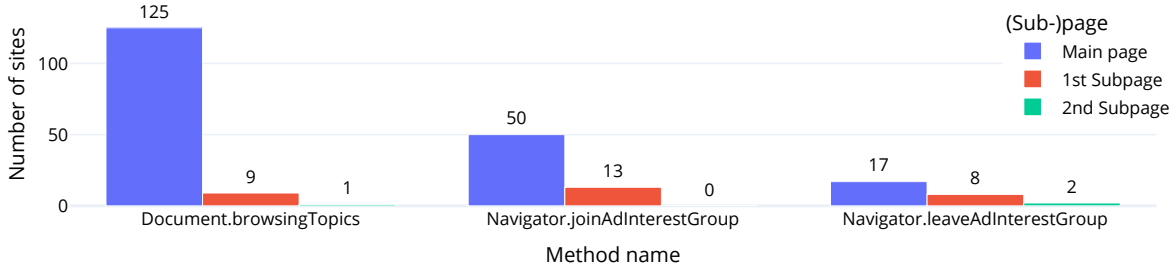

Figure 18: The number of websites on which an API call was first encountered on a certain (sub-)page (e.g. on the first subpage, but not on the main page)