Hochschule Ravensburg-Weingarten
Internet und Online-Marketing
Sommersemester 2021

Nutzer-Tracking auf Webseiten: Verbreitung, technische Umsetzung, Chancen und Risiken

Autor: Oliver Dankwart
Matrikelnummer: 31256
Professor: Prof. Dr. Marius Hofmeister

Abgabetermin: 15.07.2021

Inhaltsverzeichnis

Abkürzungsverzeichnis

AdTech Advertising Technology

AJAX Asynchronous JavaScript And XML

API Application Programming Interface

CCPA California Consumer Privacy Act

CDN Content Delivery Network

CSS Cascading Style Sheets

DOM Document Object Model

DSGVO Datenschutz-Grundverordnung

EFF Electronic Frontier Foundation

EU Europäische Union

FLoC Federated Learning of Cohorts

GAFAM Zusammensetzung der Technologie-Konzerne Google, Amazon, Facebook, Apple, Microsoft

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

ID Identifier

IP Internet Protocol

ISP Internet Service Provider

ITP Intelligent Tracking Prevention

MB Megabyte

NSA National Security Agency

URL Uniform Resource Locator

Abbildungsverzeichnis

Quellcodeverzeichnis

1 Einleitung

In den letzten Jahren ist die Online-Privatsphäre und die Gefahren des Trackings immer mehr in den Fokus der Nutzer von Online-Services gerückt. Dies ist unter anderem daran zu erkennen, dass schon 2017 im Durchschnitt 71 % der Bürger der europäischen Union (EU) über Cookies und den damit verbundenen Tracking-Möglichkeiten Bescheid wussten und auch 33 % der EU-Bürger Cookie-Tracking weitestgehend im Browser einschränken [10]. Das alles bevor die EU-weite Datenschutz-Grundverordnung (DSGVO) am 25. Mai 2018 verabschiedet wurde und in Kraft getreten ist.

Wo es noch früher rechtlich möglich war über die IP-Adresse einzelne User einer Website ausfindig zu machen, haben sich die Machtgefüge durch die EU-weite Verordnung oder aber auch dem California Consumer Privacy Act (CCPA) in Richtung der User verschoben. Denn IP-Tracking ist rechtlich nun zwecks personenbezogener Daten äußerst kritisch [11]. Dies scheint zwar wie ein vermeintlicher Gewinn für den einzelnen User auszusehen, doch werden die großen Technologie-Unternehmen wie Google & Co. andere Möglichkeiten finden ihre User eindeutig zu tracken, was schon derzeit u.a. von der oben genannten Technologie-Firma mit Hilfe von speziellen User-ID´s oder Gruppen-ID‘s getestet wird. Der Anstieg des Trackings und das wachsende Quasimonopol dieser Unternehmen sind so kaum zu verhindern.

Auf der anderen Seite ist das Tracking von Nutzern essenziell für gewisse Branchen. In Bereichen wie E-Commerce, Online Advertising und Web Analytics ist das Tracking von großer Bedeutung und das Fehlen der Tracking-Möglichkeiten würde die Qualität der Services dramatisch verschlechtern [12]. Aus diesem Grund scheint das Nutzer-Tracking auf Webseiten nicht einfach so ersetzbar zu sein und es müssen Möglichkeiten gefunden werden, um Web Tracking für beide Seiten sicher und transparent zu gestalten.

2 Problembeschreibung und Zielsetzung

Durch diese Projektarbeit soll aufgezeigt werden, was Nutzer-Tracking auf Webseiten ist und wie es heutzutage eingesetzt werden kann. Es soll verdeutlicht werden, aus welchem Grund es nötig sein kann Nutzer zu tracken, aber auch welche Risiken damit verbunden sind. Dazu soll dargestellt werden, was Regierungen gegen Nutzer-Tracking und für Online-Privatsphäre unternehmen.

2.1 Grenzen der Projektarbeit

Nicht in dieser Projektarbeit behandelt werden:

- Tracking-Methoden, welche zwecks veralteter Techniken nicht mehr von Bedeutungen sind (z.B. Adobe Flash-Cookies, Microsoft Silverlight Storage, etc.)
- das stark mit dem Nutzer-Tracking auf Webseiten verwandte Mobile-Tracking

2.2 Vorgehen der Untersuchung

Im Kapitel 3 sollen zu erst die Grundlagen zu Nutzer-Tracking auf Webseiten erarbeitet werden. Das bedeutet es wird das Nutzer-Tracking auf Webseiten definiert und daraufhin Bereiche aufgezeigt, in welchen dieses benötigt wird. Im Anschluss werden dann der heutige Stand, die Verbreitung und aktuelle Entwicklungen beim Thema Nutzer-Tracking dargestellt und eingeordnet. Hierbei werden auch staatliche Reaktionen und rechtliche Entwicklungen behandelt.
Das Kapitel 4 soll im nächsten Schritt dazu dienen, die verschiedenen Tracking-Arten vorzustellen und ihre technische Umsetzung, sowohl grafisch aufgezeigt, so wie mit Code-Beispielen beschrieben, zu erläutern. Dabei werden die Tracking-Arten in die drei Bereiche **serverseitiges Tracking**, **clientseitiges Tracking** und **andere Tracking-Arten** aufgeteilt und nacheinander vorgestellt.
Darauffolgend werden im Kapitel 5 die Chancen und Risiken bei der Nutzung von Nutzer-Tracking erarbeitet und bewertend gegenübergestellt.
Das Kapitel 6 schließt dann mit einem Fazit zu Nutzer-Tracking auf Webseiten und dessen Dimensionen ab.

3 Grundlagen

3.1 Definition von Nutzer-Tracking auf Webseiten

Zunächst ist es wichtig eine allgemeine Definition für das Nutzer-Tracking auf Webseiten aufzustellen. Dabei gehen die Meinungen durchweg in verschiedene Richtungen, da dieses Themenfeld recht neu ist, sich ständig weiterentwickelt und auch keine offizielle institutionelle Definition von Nutzer-Tracking auf Webseiten vorhanden ist. Daher definiert jeder Autor selbst, was Nutzer-Tracking auf Webseiten zu bedeuten hat.

Um den Begriff besser definieren zu können, sollte dieser erst einmal in seine Einzelteile zerlegt werden. Auf der einen Seite steht das Tracking. Das Wort Tracking stammt vom

Englischen Verb *to track* ab und wird unter anderem mit *Verfolgung* oder *Lokalisierung* übersetzt [13]. Dazu kann man das Wort mit verschiedenen Zusammenhängen definieren. So definiert Prof. Dr. Klaus Wübbenhorst im Gabler Wirtschaftslexikon das Tracking anhand dessen Nutzung in der Marktforschung. Dort beschreibt das Tracking die „regelmäßig wiederkehrende Untersuchung desselben Sachverhalts“ [14]. Auch wird das Wort in Verbindung mit der Sendungsverfolgung in der Logistik verwendet [14].

In Verbindung von Tracking mit den Wörtern *Nutzer* und *Website* sind diese Definitionen aber schwer anwendbar und es muss eine alternative Erklärung gefunden werden. Dazu stellt der Suchmaschinenoptimierungs-Software-Entwickler Ryte in ihrem Wiki eine im Zusammenhang passendere Definition. Daher soll Tracking im Online Marketing für das aufzeichnen und auswerten von Nutzer-Verhalten stehen. Dabei sollen verschiedenen Möglichkeiten genutzt werden, um die Nutzer Verfolgen zu können. Dazu zählt unter anderem das Nutzen von JavaScript-Code und das Setzen von Cookies im Browser des Nutzers [15]. Diese Definition deckt sich auch mit der aufgestellten Definition des Internetrechts-Portals eRecht24. Auch hier wird das Tracking als Erhebung und Auswertung von Nutzer-Verhaltens beschrieben. Dabei werden im Anschluss auch die Ziele des Trackings näher beleuchtet, welche von der zielgruppengerechteren Darstellung der Angebote auf einer Website, der Analyse von Klickpfaden, bis hin zur Bestimmung von Präferenzen in puncto Hersteller oder Produkt reichen [11]. Daraus kann man schließen, dass sich das Tracking meist nicht nur auf eine einzelne Sache konzentriert. Es müssen verschiedene sogenannte *Key Performance Indicator* in unterschiedlichen Kombinationen nachverfolgt und im Anschluss gespeichert werden. Diese sind per Definition die letztendlichen Kennzahlen, wie sie aus der Betriebswirtschaftslehre als Erfolgs- und Leistungsindikatoren bekannt sind, welche dabei helfen den Nutzer zu analysieren [16].

Auf der anderen Seite steht der Nutzer. Diese auch als User, Benutzer oder Internetnutzer bezeichnete Person kann zum einen als ein gewöhnlicher Nutzer eines Computers und des Internets bezeichnet werden oder als Mitglied eines sozialen Netzwerks oder einer Internet-Community [17]. Der Nutzer spielt hier eine tragende Rolle, denn dieser ist die nachverfolgte Person, die ein Unternehmen oder ein Website-Betreiber zwecks der oben genannten Zielen analysieren und überprüfen möchte. Er ist einer der zwei Akteure beim Tracking. Neben dem Nutzer steht nur noch die Website beziehungsweise die Web-Applikation, auf die der Nutzer zugreift [18].
Die angesprochene Website ist dann der dritte Teil des zu definierenden Begriffes. Die Sammlung aller *Hypertext Markup Language*-Seiten (HTML), welche im Internet von

einem Unternehmen oder einer Person zur Verfügung gestellt werden, wird den meisten ein Begriff sein, daher ist eine ausführliche Definition nicht von Bedeutung [19].
Wichtiger ist hervorzuheben, wie unterschiedlich das Nutzer-Tracking auf Webseiten bezeichnet wird. Dadurch, dass keine allgemein anerkannte Definition und auch Benennung des Vorgehens vorhanden ist, bezeichnet jeder Autor das Nutzer-Tracking auf Webseiten anders. In der gängigen englischen wissenschaftlichen Literatur werden meist die Begriffe *Web Tracking* ([20], [21]) und *Online Tracking* ([18], [22]) als Synonyme angewendet. Auch *User Tracking* wird in manchen Quellen angewendet [23].
Definitionen im Internet nutzen zudem auch Synonyme wie *Web Controlling*, *Web Analyse* oder auch *Traffic Analyse* [24]. Diese Synonyme stammen aber eher vom zum Web Tracking naheliegenden Fachbereich *Web Analytics*. Dieses beschreibt die Analyse und Optimierung einer Website und ist daher stärker auf die Auswertung der Daten konzentriert. Dazu ist der Begriff der Web Analytics auf den Website-Bereich sehr eingegrenzt. Aus diesem Grund zählte die Web Analytics in der Vergangenheit nur als Teilbereich zum übergreifenden Begriff *Digital Analytics*, welche die kanalübergreifende Messung, Analyse und Auswertung von Daten zur Optimierung von verschiedenen digitalen Angeboten beschreibt. Das heißt Digital Analytics ist nicht nur auf die Website einer Unternehmung eingeschränkt. Teilweise wurde schon früher der Begriff Web Analytics mit kanalübergreifenden Analytics gleichsetzt, weshalb man auch heute beide Begriffe als Synonyme im weitestgehenden Sinne nutzen kann [1].
Wie oben schon angesprochen ähneln sich Web Tracking und Web Analytics stark, die Web Analytics ist nur etwas mehr auf die Auswertung der gesammelten Daten fokussiert, während das Web Tracking eher das Daten sammeln behandelt, jedoch wie in der Definition beschrieben die Analyse dieser auch als Teilgebiet ansieht. Dabei wird Web Analytics in drei Funktionsbereiche aufgeteilt, die somit das Web Tracking noch einmal detaillierter beschreiben und die Aufgaben eingrenzen. Diese sind zum einen die Datensammlung auf der Website, die Datenspeicherung und -verarbeitung und die Auswertung der gesammelten Daten [1]. In genau dieser Reihenfolge werden nun eben erst die benötigten Daten gesammelt, diese daraufhin gespeichert und verarbeitet, um dann im Anschluss in die Auswertung dieser Daten hinüberzugehen (siehe Abbildung 3.1).

Da die Speicherung und Auswertung der Daten zwar Teil des Web Trackings sind, wird im weiteren Verlauf das Daten sammeln hervorgehoben und die später im Prozess angesiedelten Kernaufgaben vernachlässigt. Dabei muss verdeutlicht werden, dass der Datensammlungsschritt wohl der wichtigste Schritt im Prozess ist. Marco Hassler begründet dies mit dem *Garbage in, Garbage Out-Prinzip* aus der Informationstechnik. Dieses be-

Abbildung 3.1: Funktionsbereiche in Analytics-Systemen bzw. im Web Tracking, auf Basis von [1]

schreibt, dass wenn nur Müll in ein System gespielt wird, auch nur Müll am Ende herauskommen kann [1]. Daher ist es von großer Bedeutung anfangs die richtigen Daten zu sammeln, mit welchen man dann später eine Auswertung initiieren kann. Dafür gibt es drei Arten, wie Daten gesammelt und Nutzer nachverfolgt (getrackt) werden können. Dies sind die serverseitige Datensammlung bzw. das serverseitige Tracking, die clientseitige Datensammlung (clientseitiges Tracking) und alternative Methoden und Tracking-Arten, welche auch Kombinationen aus den beiden vorangegangenen sein können. Diese drei Arten sind die Überbegriffe für verschiedene Datensammlungs- und Tracking-Methoden, welche heutzutage in der Praxis angewendet werden. In den folgenden Kapiteln werden diese Tracking-Arten noch detaillierter beschrieben und ihre Funktionsweise erklärt.

Im Diskurs dazu steht meist auch der Begriff *Third-Party-Tracking*. Hierbei handelt sich um Tracker, die von Dritten auf einer Website, die ein Nutzer besucht, platziert werden, um deren Daten zu sammeln und für ihre eigenen Geschäfte nutzen zu können [21]. Dabei hat der Nutzer die Website des Dritten noch nicht einmal besucht und wird trotzdem von dessen Betreibern getrackt [25]. Im Gegensatz dazu stehen die First-Party-Tracker, die direkt vom Webseiten-Betreiber und dessen Web-Domain stammen.

Aus diesem Grund zählt das Third-Party-Tracking zu den umstrittenen Praktiken im Web Tracking-Bereich und ist einer der Gründe, warum Web Tracking einen schlechten Ruf hat. Das Third-Party-Tracking gilt als große Gefahr für die Privatsphäre der Nutzer, da die Verbreitung dieser Third-Party-Tracker über mehrere Website mittlerweile enorm groß ist. Eine Studie hat ergeben, dass auf etwa 46 % der Hauptseiten von Alexa Global Top 10.000 Websites mindestens ein Third-Party-Tracker platziert ist und dieser Daten an Dritte weiterreicht. Allein Google ist dabei bei 25 % dieser Seiten vertreten und sammelt Daten von Nutzern [26]. Neben dem klassischen Setzen eines Third-Party-Trackers mit Hilfe von Cookies direkt von einer anderen Domain (bspw. *doubleclick.net*) aus, gibt es drei weiterer Arten von Third-Party-Trackern. Auch können einfache First-Party-Cookies über Third-Party-JavaScript-Code gesetzt werden oder man nutzt First- und Third-Party-Cookies, um dann mit Hilfe eines JavaScript-Codes Nutzer nachzuverfolgen. Eine letzte Art von Third-Party-Trackern wird bei Werbung genutzt, um das Tracking über einen anderen Service durchführen zu können [21].

3.2 Bereiche für die Nutzung

Insgesamt gibt es viele unterschiedliche Bereiche, in welchen Web Tracking angewendet wird, um Nutzer online, nachverfolgen zu können und allgemein Daten zu sammeln. „Data is the new oil“ [27]. 2006 prägte der britische Mathematiker und Entrepreneur Clive Humby eine noch junge Disziplin mit diesem Zitat. Es sollte stetig wachsende Wichtigkeit von Daten für Unternehmen aufzeigen und die Zukunft in gewisser Weise bestimmen. Heutzutage basieren die meisten Entscheidung auf Daten bzw. Nutzer-Daten, die vorher gesammelt werden mussten. Ohne diese würde Google mehr als 120 Milliarden Dollar pro Jahr an Werbeeinnahmen verlieren [28] oder Facebook ihre insgesamt über vier Milliarden Nutzer auf Facebook und Instagram nicht mit gezielter und personalisierter Werbung ansprechen [29]. Wegen unter anderem diesen Unmengen an Daten sind die großen Technologie-Unternehmen eben so mächtig und diese Daten mussten und müssen eben erst gesammelt werden um sie nutzen zu können. Und hier kommt das Web Tracking in Erscheinung und zeigt, wie wichtig es in verschiedenen Bereichen ist.

Online Advertising

Allen voran wird das Web Tracking für das *Online Advertising* genutzt. Letztendlich wurde Web Tracking dafür entwickelt Marketing einfacher zu gestalten und Umsätze zu erhöhen [21]. Online Advertising ist die Schaltung von Werbung im Internet, welche auf Desktop-, wie auf Mobil-Geräten dem Kunden ausgespielt werden. Zu den verschie-

denen Online Advertising-Arten zählen unter anderem das *Banner Advertising*, wie es von Nachrichtenseiten wie *t-online.de* als seitliche Werbebanner bekannt ist, das *Video Advertising* (z.B. YouTube), die *Suchmaschinenwerbung* (Google Suchanzeigen) und das *Social Media Advertising* (z.B. Werbung auf Instagram, Facebook & Co.) [30].
Online-Werbung wird von Jahr zu Jahr immer wichtiger. 2019 hatte Online-Werbung 11,6 % Marktanteil unter den Werbemedien in Deutschland (siehe Abbildung 3.2) [2], was zwar in Anbetracht des Monopols der Fernsehwerbung als unwichtig erscheinen mag, doch sieht man sich einmal Prognosen zu den Werbeausgaben pro Werbemedium in den nächsten Jahren an, kann man erkennen, dass weiterhin viel mehr in die Internet-Werbung investiert wird (32,46 % in Paid Search bis 2022) und Fernsehen (-17,51 %) immer weiter unwichtiger wird [31]. Online Advertising wird somit also weiterhin wichtig bleiben und sogar noch essenzieller werden. Damit Online Advertising effizient abläuft benötigt es das Web Tracking. Nur durch das Web Tracking und die Nutzung der gesammelten Daten konnten Werbenetzwerke, wie diese von Google oder Facebook, in der Vergangenheit das Wissen der Daten nutzen, um Werbung von Werbetreibenden zielgruppenspezifisch in verschiedenen Formen an die jeweiligen Nutzer auszuspielen und selbst als Monopolisten im Werbemarkt aufzusteigen.

Dazu ist auch das Affiliate Marketing ein Teilbereich des Online Advertisings. Dabei handelt es sich um ein internetbasiertes Provisionssystem, bei welchem ein Website-Betreiber als sogenannter *Affiliate* die Produkte einer Marke bzw. eines Unternehmens bewirbt. Dafür erhält der Website-Betreibende unter verschiedenen Bedingungen, meist aber nach jedem erfolgreichen angebahnten Verkauf an einen Kunden, eine Provision [32]. Damit dies aber funktioniert, muss der potenzielle Kunden schon auf der Website des Website-Betreibenden bis hin zur Abschlussseite des schlussendlichen Kaufes durch Web Tracking nachverfolgt werden. Nur so kann eine genaue und fehlerfreie Auszahlung der Provision an den Affiliate garantiert werden [21].

E-Commerce

Ein weiterer Bereich, in welchem das Web Tracking nötig ist, ist die *E-Commerce-Branche*. E-Commerce-Websites wie Amazon oder Zalando sind darauf spezialisiert die gesammelten Daten optimal zu nutzen, um daraus einen Nutzen zu ziehen. Sie tracken dabei die Nutzer, um die Website zu optimieren und mehr Produkte verkaufen zu können. In Analysen beantworten sie fragen, wie „Wie lange wurde ”gebrowst”, bevor das Produkt gekauft wurde?“, „Wann wurde nach welchem Produkt gesucht?“ oder „Wurde ein Produkt in den Warenkorb gelegt und dann gekauft oder danach der Kaufprozess

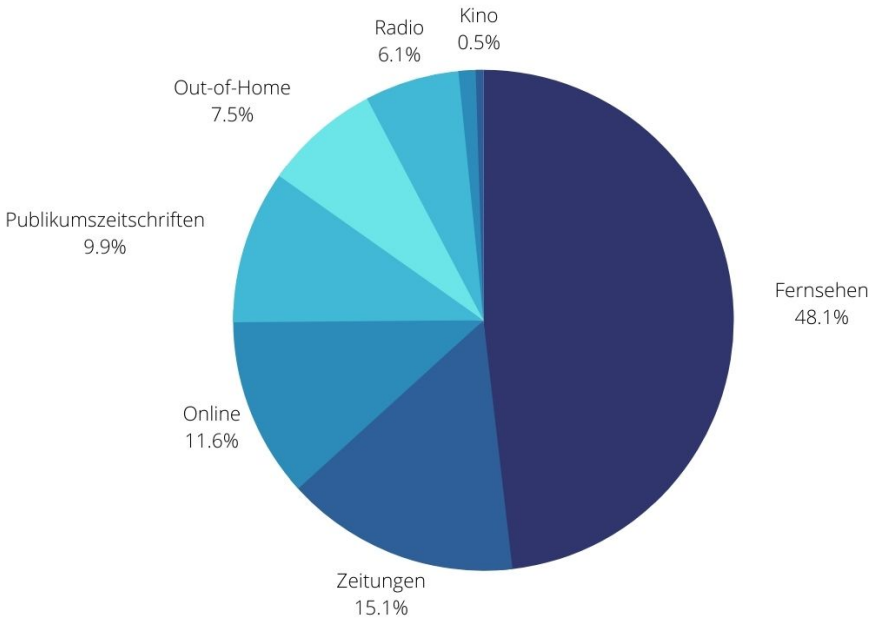

Abbildung 3.2: Marktanteile der einzelnen Werbemedien im deutschen Bruttowerbemarkt im Jahr 2019, auf Basis von [2]

abgebrochen?“. Die E-Commerce-Websites versuchen dadurch die Gewohnheiten und Präferenzen ihrer Nutzer zu lernen und zu verstehen, um daraus dann Profit schlagen zu können [12].

Eine weitere Frage die bei den Analysen der E-Commerce-Website-Betreiber auch häufig gestellt wird ist „Hat der Nutzer ein Produkt zu einem bestimmten Preis gekauft, aber nicht zu einem anderen?“ [12] Häufig werden Preise auf kommerziellen Websites anhand verschiedenen Datenpunkten, die über den Nutzer gesammelt wurden, angepasst, so dass diese potenziell mehr ausgeben, wenn erkannt wird, dass ein Nutzer eventuell durch seiner Herkunft, Alter oder anderen Daten mehr Geld ausgeben würde. Diese Praktik kann zum Überbegriff Preisdiskriminierung bzw. Preisdifferenzierung gezählt werden. In der Vergangenheit wurden schon einige Studien in verschiedenen Branchen durchgeführt, die bewiesen haben, dass diese Methode oft angewendet wird. Zum Beispiel wurde in einer Studie bewiesen, dass der Preis anhand dem geographischen Standort, dem Wohlstand und dem Referer (die Website von welcher der Nutzer zur Zielseite weitergeleitet wurde) angepasst wird [33]. In der Finanzbranche hat die Chase Bank die Chase Sapphire Card mit verschiedenen Zinsraten angeboten und dabei Rechenalgorithmen eines Dienstleister genutzt, welcher auf verschiedene Nutzer-Daten, wie z.B. der Postleitzahl

oder das Geburtsdatum, zurückgegriffen hat, um die Zinsraten zu verändern [21]. Auch in der Hotel-Branche wird Web Tracking zur Optimierung der Preise angewendet. So werden höherpreisige Hotels an Mac-Nutzer als Werbung an diese ausgespielt und die günstigeren Hotels an die Windows-Nutzer [21].

Web Analytics

Wie schon oben ausführlich erklärt wird Web Tracking auch enorm in der *Web Analytics* genutzt. Web Analytics beschreibt den ganzen Prozess von der Datensammlung, über die Speicherung und Verarbeitung, bis hin zur Auswertung in Analysen. Hier ist es essenziell im ersten Schritt mit Hilfe von Web Tracking die richtigen Daten zu sammeln, welche dann bei der Auswertung genutzt werden, um Geschäftsentscheidungen treffen zu können. Heutzutage ist es von enormer Wichtigkeit einen erfolgreichen Webauftritt zu besitzen, welcher die angebotenen Produkte und Leistungen attraktiv präsentiert und die Kontaktaufnahme mit dem Unternehmen garantiert, damit die Produkte und Leistungen letztendlich verkauft werden können [34]. Daher muss die Website immer wieder analysiert und auf den Ergebnissen basierend verbessert und optimiert werden. Web-Admins lernen durch Web Analytics, wie Nutzer die Website nutzen, was diese gut oder schlecht finden und wer überhaupt die Nutzer sind [12]. Im gleichen Zusammenhang wird das Web Tracking auch für das *Usability Testing* eingesetzt. Darunter versteht man im Web-Bereich einen Prozess, in welchem Nutzer eine Website nutzen und testen, während dabei deren Bewegungen auf der Website aufgezeichnet und dann für spätere Optimierungen auf der Website genutzt werden können [21]. Dies kann zum einen in einem Laborversuch mit einer klar definierten Aufgabe für den Testprobanden oder in einem Feldexperiment direkt auf der Live-geschalteten Website durchgeführt werden. Eine Studie hat gezeigt, dass von den Top 1.300 Websites im Alexa Ranking 115 Websites des Tastatur- und Maus-Trackings verdächtig waren. Dabei nutzte eine handvoll dieser Websites angefügte Parameter im Uniform Resource Locator (URL), um Klicks zu tracken und diese Informationen an einen Drittanbieter weiterzureichen [35]. Nichtsdestotrotz gilt aber die Web Analytics und das Usability Testing im Normalfall als keine Gefahr für den Nutzer und dessen Privatsphäre, da die Daten meist nur auf der eigenen Herkunftswebseite genutzt und verarbeitet werden [21].

Regierungen und staatliche Geheimdienste

Zuletzt kann das Web Tracking auch für *Regierungen und staatliche Geheimdienste* nützlich sein. Dabei kann Web Tracking angewendet werden, um bestimmte Personen zu

überwachen und verdächtigte Straftäter zu überführen. Das Thema staatliche Überwachung ist schon seit längerer Zeit im Gespräch, kochte aber vor allem durch die Enthüllungen vom US-amerikanischen Whistleblower Edward Snowden nochmals weiter hoch. Dort wurden Überwachungsnetzwerke und deren Methoden veröffentlicht, die eigentlich unter Geheimhaltung standen [36]. Die National Security Agency (NSA) arbeitete unter anderem mit Google zusammen, um mit dessen Tracking-Methoden Personen Online ausfindig machen zu können und nutzten IP-Adressen um Standorte herauszufinden [21]. Auch wurde aufgedeckt, dass die Regierung der Vereinigten Staaten im Zeitraum vom Januar 2014 bis zum Juni 2014 12.539 Anfragen für 21.576 Personen und deren personenbezogenen Daten inklusive Suchverlauf an Google gestellt hat und Google bei 84 % der Fälle kooperiert hat [21]. Nicht nur offensiv wird Web Tracking für die Staatsüberwachung genutzt, sondern auch in passiver Natur. So fand die WhoTracks.me-Initiative, welches ein Zusammenschluss des ehemaligen Webbrowser- und Browsererweiterungs-Entwickler Cliqz und des Werbeblocker-Entwicklers Ghostery ist, heraus, dass selbst auf den Regierungsseiten von manchen Staaten Third-Party-Tracker von Google. & Co. genutzt werden, welche Daten über den Nutzer sammeln können und das selbst ohne das der Nutzer einwilligen muss [3]. In dieser Statistik fallen vor allem die Regierungsseiten der Vereinigten Staaten negativ auf, welche mit etwas mehr als 2,4 Trackern pro Regierungsseite die Spitze darstellen (siehe Abbildung 3.3). Interessanterweise konnten bei der Studie keine Third-Party-Tracker auf deutschen Regierungsseiten aufgespürt werden.

Insgesamt fällt die deutsche Regierung bei der Nutzung von Web Tracking nicht wesentlich stärker auf wie andere Staaten. So ist es bekannt, dass der Bundesnachrichtendienst einen der größten Internetknoten (DE-CIX in Frankfurt) beobachtet und auch Daten davon abgegriffen werden [37]. Zudem wurde auch veröffentlicht, dass das Bundeskriminalamt die Möglichkeit besitzt Nachrichten aus WhatsApp-Konten in Echtzeit zu beobachten [38]. Auch steigen die Anfragen des Bundeskriminalamts an Internetanbieter nach Daten über deren Nutzer [39]. Jedoch im Großen und Ganzen sind die Informationen aus der NSA-Affäre über die regelmäßig Weiterreichung enorm vieler Datenpunkte an die NSA das einzige Indiz für ein negatives Bild der deutschen Regierung im Zusammenhang mit Datenmissbrauch [40].

Anreicherung von Daten

Als weitere wichtige Nennung ist das Thema der Nutzung von Web Tracking als Anreicherung der schon bestehenden Daten zur Optimierung von Diensten außerhalb des Web-Bereichs. So nutzen manche Finanz-Unternehmen die Online-Aktivitäten, um die

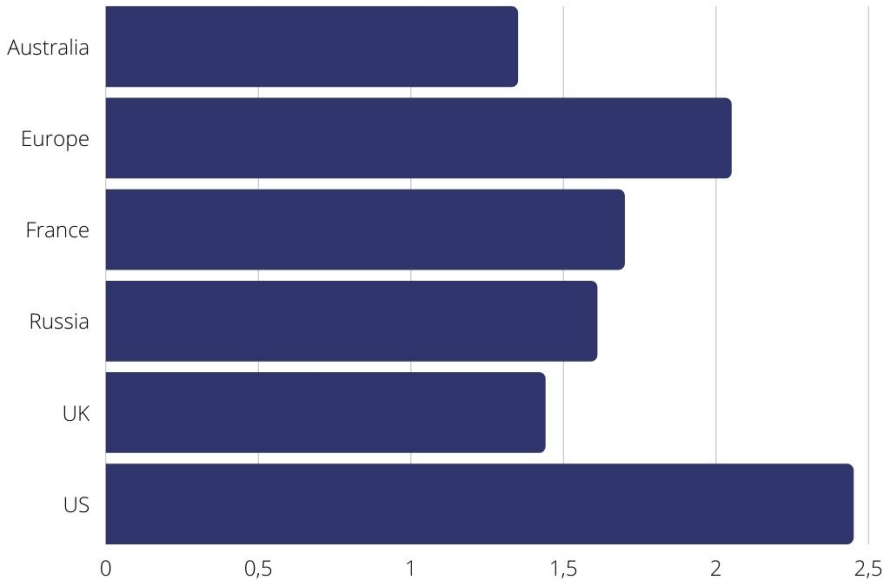

Abbildung 3.3: Durchschnittliche Anzahl an Trackern auf ausgewählten Regierungsseiten im September 2018, auf Basis von [3]

Kreditwürdigkeit von Personen festzustellen. So hat Deutschlands führende Auskunftei für Kreditwürdigkeit 2012 geplant Daten von Facebook-, LinkedIn- und Twitter-Accounts zu sammeln, um die Vernetzung zwischen verschiedenen Personen ausfindig zu machen [21]. Solche Methoden werden auch in der Versicherungs-Branche angewendet, um anhand von Daten zu bestimmen, welche Versicherung unter welchen Konditionen einer Person angeboten werden sollten [21].

3.3 Status quo, Verbreitung und aktuelle Entwicklungen

Das Web Tracking hat in den letzten Jahren für viel Aufsehen gesorgt. Dies geschah vor allem in Verbindung mit dem Thema Online-Privatsphäre. Auch wenn dieses Thema schon in den 90er-Jahren während der Hochzeit von Identitätsdiebstahl über das Internet immer wichtiger für den einzelnen wurde, nahm das Verlangen nach mehr Online-Privatsphäre in den 2000er- und 2010er-Jahren stark zu [41]. Nicht allein durch die angesprochenen Veröffentlichungen Edward Snowdens zu staatlichen Überwachungsprogrammen. Auch durch das rasante Wachstum von Technologie-Giganten, wie Facebook und Google und die eingesetzten Mittel, um Unmengen an Daten über deren Nutzer sammeln zu können, zeigt wie wichtig es ist heutzutage Online-Privatsphäre nicht hin-

ten anzustellen.

Wie verbreitet ist Web Tracking?

Web Tracking ist dabei ein zentraler Baustein, welcher bestimmt, wie und welche Daten über einen Nutzer gesammelt werden. Um die Bedeutung dessen zu verdeutlichen muss gezeigt werden, wie verbreitet das Web Tracking ist. Eine Studie von WhoTracks.me in 2019 hat ergeben das auf Basis von 340 Millionen besuchten Webseiten im April 2018 71 % dieser Seiten mit einem Tracker versehen sind. Dabei wurde eine durchschnittliche Anzahl an Trackern pro Webseite von acht errechnet. Im Zuge dessen ist auch hervorzuheben, dass Google Analytics allein auf 46 % der besuchten Seiten vertreten war und Google selbst acht mal in der Top 10 der Third-Party-Tracker vorgekommen ist, auch wenn Google Fonts und Google APIs nicht im Tracking-Kontext genutzt werden, also keine heiklen Daten weiterreichen (siehe Abbildung 3.4). Google war insgesamt als Unternehmen auf 82 % der Webseiten im Sinne eines Third-Party-Trackings vertreten [4]. Aus einer vorangegangenen Studie des WhoTracks.me-Mitglied Ghostery wurde im Jahre 2017 bekannt, dass bei 77,4 % aller geladenen Webseiten Tracker implementiert sind [42], weshalb man meinen könnte, dass die Anzahl an Trackern über die Jahre sank. Ein Grund dafür könnte sein, dass durch die Einführung der DSGVO und den damit verbundenen Veränderungen in der Online-Advertising-Branche diese Kennzahl beeinflusst wurde. Nichtsdestotrotz zeigen die Studien, dass die Verbreitung von Online-Tracking extrem hoch ist. Näherungsweise kann man sagen, dass von vier Webseiten fast drei Web Tracking nutzen. Zieht man dazu noch die schon angesprochene Studie zum Third-Party-Tracking dazu, welche beschreibt, dass bei 46 % der Alexa Top 10.000 Webseiten Tracker genutzt werden [26], erkennt man schnell, dass das Web Tracking, ob First- oder Third-Party, eine weit verbreitete Methodik ist.

Auch die Tageszeitung *The New York Times* befasste sich im Zuge einer Recherche für einen Tracking-Artikel mit dem Tracking im Internet. Dabei wurde ein Redakteur für einen Arbeitstag beobachtet und danach dessen Tracking-Daten ausgewertet. Zusammengefasst wurde der Redakteur beim Besuch von 47 Webseiten über den ganzen Tag hinweg von hunderten Trackern verfolgt. Es wurden unter anderem der geographische Ort, die Region und somit dessen politische Wichtigkeit im Sinne der sogenannten ”Swing States” bei den Wahlen in den Vereinigten Staaten und auch einige Browser-Details getrackt und gesammelt [43].

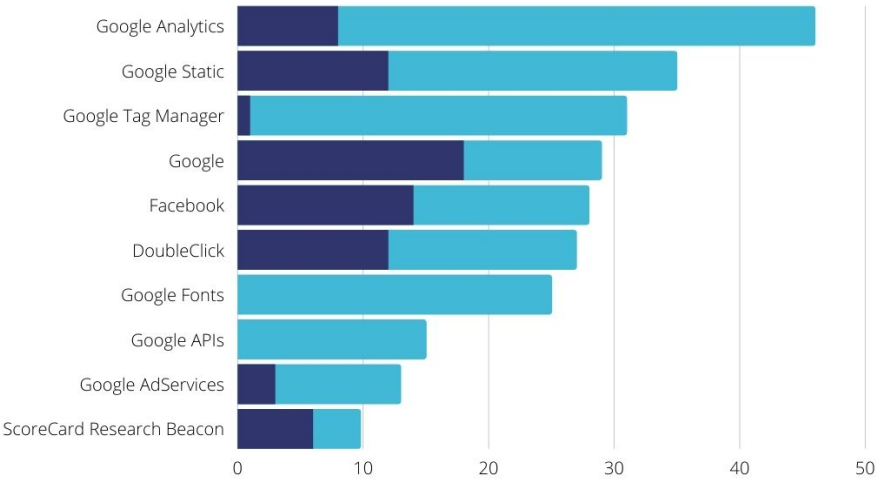

Abbildung 3.4: Top 10 Third-Party-Trackers nach Verbreitung, auf Basis von [4]

Wie gefährlich kann Web Tracking sein?

Wie gefährlich diese exzessive Datensammlung sein kann, zeigen vielerlei Vorfälle bei Facebook. Nicht erst zuletzt ist Facebook mit dem Datenleak von mehr als 500 Millionen Facebook-Nutzern abermals in Kritik geraten. Neben Facebook-Nutzernamen wurden auch der vollständige Name, Telefonnummer, Geburtsdatum, geographischer Ort, biografische Angaben und teilweise auch die E-Mail-Adresse veröffentlicht [44]. Diese Daten sollen aber von einem alten Leak aus dem Jahr 2019 stammen, wobei hier anzumerken ist, dass der besagte Leak im August 2019 auch Facebook-IDs von Nutzern veröffentlichte. Dieser Identifier (ID) ist eine einzigartige lange Zahl, welche jeden Facebook-User unterscheiden soll [45]. Mit diesen Daten können Hacker einfache Angriffe auf diese Facebook-Nutzer initiieren und gegebenenfalls mehr Schaden, auch auf finanzieller Basis, anrichten.
Dies ist nur ein Beispiel für Vorfälle bezüglich Datensammlung und auch wenn noch keine Facebook-Werbeprofile, die durch Web Tracking erstellt wurden sind, veröffentlicht wurden, scheint es doch eine bestehende Gefahr dessen zu geben. Dabei ist aber der große Unterschied hervorzuheben, dass die Erstellung eines Profils bei Facebook im direkten Interesse des Facebook-Nutzers steht, da dieser nun mal sich dazu entscheidet ein

Facebook-Konto zu erstellen und dabei Daten preiszugeben. Hier werden bewusst Daten weitergegeben. Ein Klick auf eine Google-Anzeige oder der Besuch einer Website ist aber keine Entscheidung bei der ein direktes Interesse beim Kunden besteht, Daten preisgeben zu wollen. Hierbei wird der Nutzer auch nicht aktiv davor mit der Sammlung von Daten nach dem Klick auf eine Anzeige oder Website konfrontiert. Dies hat sich vor allem in der EU nach der Einführung der *DSGVO* geändert, da nun Website-Besucher vor der Interaktion mit der Website gewarnt werden, dass Daten gesammelt werden und dazu noch die Auswahl besteht, welche Daten gesammelt werden dürfen. Dies alles geschieht im Rahmen der Cookie-Meldungen, meist am Rand auf den Webseiten dargestellt. Die DSGVO und weitere staatliche Reaktionen werden im Anschluss behandelt.

Web Tracking Status Quo

Zunächst ist es von Wichtigkeit die populären Web Tracking-Arten in Kürze darzustellen. Die wohl bekannteste Art ist das Tracking über Cookies, welche im späteren Verlauf näher erläutert wird. Das liegt vor allem an der Online Advertising-Branche und das Third-Party-Tracking, welches schon erläutert wurde. Mittlerweile steht diese Technologie aber kurz vor dem Ende. Ohne Third-Party-Tracking bzw. der Nutzung von Third-Party-Cookies wird eben das effiziente Online Advertising schwierig und ist daher von äußerster Wichtigkeit. Seit einigen Jahren blockieren die gängigen Browser Third-Party-Tracking. Der Mozilla Firefox und der Safari Browser haben schon seit 2013 die Funktion Third-Party-Cookies zu blockieren [46]. Seit 2019 ist diese Funktion bei Firefox standardmäßig aktiviert [47] und auch eine Möglichkeit zur Isolierung von First-Party-Trackern wurde in einer früheren Version eingeführt, ist aber standardmäßig deaktiviert, da diese Einstellung manche Webseiten und Web-Services unbenutzbar machen und teilweise zerstören könnte [48] [49]. Safari ist bekannt für die Intelligent Tracking Prevention (ITP). Dies ist eine von Webkit, der Open-Source Web-Browser-Engine Entwickler, welcher dem Safari-Browser als Basis unterliegt, entwickelte Funktion, um First-Party-Tracking zu blockieren. Dabei wird ein Machine Learning-Modell angewendet, welches erkennen soll, welche Domains Cross-Site-Web-Tracking, also das Tracken eines Nutzers über mehrere Websites hinweg, betreiben. In den ITP Versionen 1.0 und 1.1 gab es noch ein 24-Stunden-Fenster, in welchem ein First-Party-Tracker nicht blockiert wird, wenn der Nutzer die Website besucht. Somit sollten sich eigentlich Retargeting-Möglichkeiten einschränken lassen, doch hat auch dagegen die Advertising-Technology-Branche (AdTech) eine Möglichkeit gefunden, diese Einschränkungen zu umgehen. Daraufhin stellte Apple im Juni 2018 ITP 2.0 vor und verwarf hierbei die Funktion des 24-Stunden-

Zeitfensters und blockte nun alle First-Party-Tracker. Auch folgende Versionen bis hin zur ITP 2.3 aus dem September 2019 machten es noch schwieriger für Online Advertiser First-Party-Tracking nutzen zu können [50]. Neben den gängigen Browsern sind auch neue Start-Ups dazugestoßen und entwickeln Anti-Tracking-Browser. So ist der Brave Browser als Beispiel zu nennen, welcher auf Chromium, dem Unterboden von Google's Chrome Browser, basiert. Brave bringt dabei Features, wie unter anderem Ad-, Script- und Fingerprint-Blocking, mit und reiht sich so neben Safari und Firefox als eine weitere Alternative für Anti-Tracking-Browser ein [51]. Diese drei Browser stellen aber insgesamt nur etwa 12 % des Marktanteils der Web Browser im April 2021 und sind weit abgeschlagen vom klaren Marktführer Google Chrome, welcher mit über 61 % per gängigen Definitionen als Monopol gelten kann (siehe Kartellrecht § 18 Absatz 4 Gesetz gegen Wettbewerbsbeschränkungen [52]) [53]. Google hat nun aber Anfang 2020 das Ende der Third-Party-Cookies bzw. -Trackings für 2022 im Chrome Browser angekündigt und folgt somit den Marktteilnehmern in eine "Cookielose" Welt [54].

Aus den oben gezeigten Bewegungen gegen das übermäßige Tracking mit Hilfe von Cookies sind neue Tracking-Arten aufgestiegen, die das Cookie-Tracking ablösen kann. Eine dieser neuen Tracking-Arten ist das Fingerprinting, was kurz gesagt einen digitalen Fingerabdruck des genutzten Browsers macht. In einer Studie konnte herausgefunden werden, dass auf mehr als 25 % der Alexa Top 10.000 Websites Fingerprinting eingesetzt wird, vor allem bei News-Seiten, welche meist viel Werbung zwecks Monetarisierung schalten müssen [55]. Mittlerweile gibt es aber wie oben schon erwähnt Browser, welche Fingerprinting blockieren. So blockiert der Brave Browser Fingerprinting teilweise und war 2020 beim Entwickeln eines Zufallsgenerators für den Fingerprint des Browser, so dass kein eindeutiger Fingerabdruck zum Tracking genutzt werden kann [56]. Die Electronic Frontier Foundation (EFF), welches eine non-profit Organisation aus den Vereinigten Staaten für die Verteidigung von digitaler Privatsphäre, Meinungsfreiheit und Innovation ist [57], hat dafür auch ein Online-Awareness-Tool veröffentlicht, mit welchem der eigene Browser auf Fingerprinting-Gefahren geprüft werden kann und sich zum Thema informiert werden kann: `https://coveryourtracks.eff.org/`
Ein Ergebnis im Tool bei Nutzung des Brave Browser kann in Abbildung 3.5 betrachtet werden. Hier ist auch zu erkennen, dass der Brave Browser den besagten Zufallsgenerator für den Browser Fingerprint nutzt. Neben dem Brave Browser nutzen aber mittlerweile auch Firefox [58] und der Safari Browser [59] Anti-Fingerprinting-Methoden.

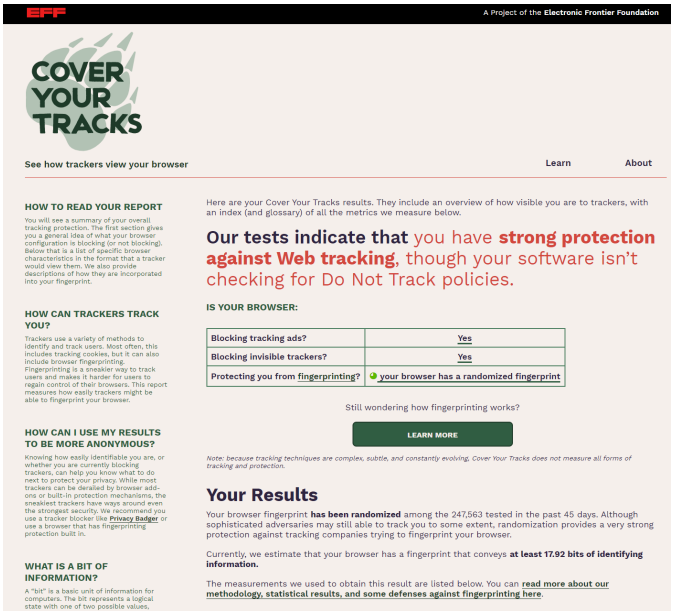

Abbildung 3.5: Ergebnis des Cover Your Tracks-Tools von EFF bei Nutzung des Brave Browsers

Aktuelle Entwicklungen

Als weitere Reaktion auf die Ankündigung des bevorstehenden Endes von Third-Party-Cookies durch den beliebtesten Browser Google Chrome hat sich das AdTech-Unternehmen *The Trade Desk* zur Aufgabe gemacht eine Alternative zum Third-Party-Tracking mit Hilfe von Cookies zu finden. The Trade Desk ist eigentlich eine Einkaufs-Plattform für Media und Werbung, aber ist zugleich mit der *Unified ID* der Begründer einer Lösung, mit welcher Cookies aus verschiedenen Quellen miteinander abgeglichen werden können, um so deren Zuordnung zu Nutzern und die an diese Nutzer gerichtete Werbung effizienter zu gestalten. Mit der Unified ID Version 1.0 wird demnach nur mit Cookies gearbeitet, was in Zukunft keine Lösung mehr sein kann. Daher entschied sich The Trade Desk die Entwicklung der Unified ID 2.0 zu initiieren, welche als Open-Source-Projekt (und somit frei von monopolistischen Strukturen) betrieben wird und ganz auf Cookies verzichten soll. Die Idee ist, dass ein Nutzer sich per E-Mail-Adresse und Zusage, dass diese für Werbezwecke genutzt werden darf, sich bei einer Website anmeldet und diese E-Mail-Adresse dann unter den Publishern weitergegeben werden kann. Die jeweiligen beteiligten AdTech-Unternehmen bekommen derweil nur eine verschlüsselte ID auf Basis der E-Mail-Adresse. Insgesamt sind sich aber Kritiker nicht sicher, ob dieser E-Mail-Login vom Nutzer angenommen wird und wie Betrugsfest diese Form ist [60].

Mit dieser Lösung stehen sie nicht alleine da, auch Konkurrenten in der AdTech-Branche arbeiten an ähnlichen Lösungen, um das Cookie-Tracking abzulösen. So entwickeln Konkurrenten wie ID5 oder Liveramp, letzterer ist nun auch bei der Unified ID 2.0 beteiligt [61], login-basierte cookielose ID-Systeme [60]. Zudem entwickeln die beiden Partner Prebid und Epsilon die SharedID-Lösung, welche als Ergänzung zu den login-basierten ID-Systemen alle nicht eingeloggten User abdecken soll [62]. Mittlerweile hat nun The Trade Desk die Verantwortung für das Unified ID 2.0-Projekt an Prebid abgegeben, um Interessenkonflikte beiseite legen zu können [63]. Es wird sich in naher Zukunft nach Fertigstellung des Projekts zeigen, wie effektiv die Unified ID 2.0 sein wird und wie somit die auferlegten Schranken der gängigen Web Browser umgangen werden können.
So wie man beim Unified ID 2.0-Projekt nicht genau weiß was das Resultat bringen wird, so gab es schon ein Resultat bei den Datenschutz-Neuerungen von Apple. Ende des Jahres 2020 kündigte Apple an, es einfacher für die Apple-Nutzer zu machen Tracking zu verhindern. So soll in 2021 die Funktion hinzugefügt werden, dass Nutzer vorher abgefragt werden, bevor einer App das Tracking erlaubt wird [64]. Dafür hat Apple Kritik von unter anderem Facebook und Google geerntet, welche dazu noch die Vorhersage verfassten, dass nur wenige Nutzer diese Abfragen akzeptieren und somit Tracking unterbinden werden [65]. Dadurch würde der ganzen Werbe-Branche geschadet. Im Mai 2021 bewahrheiteten sich nun die Aussagen und eine aktuelle Studie der App-Analysten Flurry zeigt nun, dass weltweit gerade mal 15 % der Apple-User das Opt-In durchführen, in den Vereinigten Staaten allein sind es gerade mal 6 % (Stand: 16. Mai. 2021) [66]. Hier sieht man die Kehrseite der Anti-Tracking-Bewegung. Zwar versuchte hiermit Apple einen neuen Ansatz Datenschutz in die Hände des Nutzers zu geben, so schadete dies der Werbe-Branche aber enorm.

Auch wenn Google bei den neuen Datenschutzänderung von Apple Kritik ausübt, so werden sie selbst derzeit stark kritisiert. Grund dafür sind die neuen Vorschläge als Alternativen zum Third-Party-Tracking. Nachdem Google ankündigte, dass Third-Party-Tracking mit Hilfe von Cookies im Chrome Browser bis 2022 nicht mehr möglich sein werde, beschwerten sich viele aus der Werbe-Branche über diese Umstände. Man klagte dabei an, dass Google somit ihr Datenmonopol ausspiele und nun nicht mehr auf Cookies angewiesen sei, sondern über Chrome und andere Produkte genug Daten über deren Nutzer sammeln könne, um deutlich bessere personenbezogene Werbung ausspielen zu können als die Konkurrenz [67]. Daher beschloss Google im Zuge ihrer „Privacy Sandbox“-Initiative vor dem Web-Standardisierungs-Konsortium W3C einige Vorschläge vorzustellen, die den Werbemarkt und das zugehörige Tracking weiterhin sicherstel-

len sollen. Das wohl wichtigste und am stärksten im Diskurs stehende Konzept ist das "Federated Learning of Cohorts" (*FLoC*). Bei dieser Lösung werden Browseraktivitätsdaten aus dem Chrome Browser genutzt, um den Nutzer in eine Zielgruppe/Kohorte von etwa 1.000 Personen einzuteilen, welche wiederum Werber nutzen können, um die passende Werbung auszuspielen. Dadurch hat nun nicht jeder einzelne Nutzer eine einzigartige ID, sondern nur die Gruppe an Nutzern eine Gruppen-ID. Solch eine Technik nennt man „k-Anonymität" [67]. Somit ersetzt dieses Konzept den Third-Party-Cookie komplett. Kritiker sind aber schon auf FLoC aufmerksam geworden und bringen Gegenargumente. Allen voran kritisiert die EFF das neue Konzept in einem Beitrag. „The technology will avoid the privacy risks of third-party cookies, but it will create new ones in the process." [68] Weiter beschreibt die EFF, dass Fingerprinting-Methoden einfacher eingesetzt werden können, da mit einer neuen ID im Browser die Einzigartigkeit eines jeden Users in einer Kohorte von etwa 1.000 Personen durch Fingerprinting um einiges einfacher festzustellen ist. Dazu besteht die Gefahr, dass Informationen aus der FLoC ID genutzt werden können, um einen Nutzer genauer beschreiben zu können. So könnten die FLoC-Informationen mit einer ohnehin schon vorhandenen Google-Anmeldung auf einer Website oder bei einem Web-Service verknüpft werden, um ein Nutzerprofil für Marketingzwecke anzureichern. Tracker könnten Reverse-Engineering auf dem FLoC-Algorithmus anwenden, um herauszufinden, welche User zu welcher Kohorte gehört. Auch könnten Webseiten-Betreiber im allgemeinen analysieren, dass Nutzer einer Kohorte mit einer gewissen FLoC ID sich bestimmt Verhalten und einer Zielgruppe zugeordnet werden können. Das EFF kritisiert den Weg des Targetings in der Advertising-Branche im allgmeinen schon als diskriminierend ein und sieht das FLoC-Konzept in keine andere Richtung voranschreiten [68]. Google hat nun schon erste Tests initiiert, um das neue Konzept im Feldexperiment ausprobieren zu können.
Mittlerweile hat sich neben dem EFF auch das wohl am weit verbreitetste Content-Management-System WordPress gegen FLoC öffentlich ausgesprochen und bestärkt das Blocken von FLoC [69] [70]. Die Entwickler des Brave Browser haben zudem die Deaktivierung von FLoC im Browser implementiert [71], auf der Website von GitHub wurden schon Maßnahmen zur Deaktivierung gesichtet [72] und der Suchmaschinen-Entwickler DuckDuckGo entwickelte kurzerhand eine Chrome-Erweiterung, welche FLoC blocken soll [73]. Dies sind nur einige wenige Beispiele von Gegenmaßnahmen zu FLoC. Ein Tool der Electronic Frontier Foundation hilft dabei zu erkennen, ob im eigenen Chrome Browser der FLoC-Test von Google schon ausgerollt wurde: `https://amifloced.org/`.

Staatliche Reaktionen und rechtliche Veränderungen

Auf der staatlichen Seite wird auch versucht die Kontrolle über das Web Tracking und die Datensammlung zu übernehmen. Allen voran steht dabei die DSGVO. Die DSGVO (engl. General Data Protection Regulation - GDPR) ist eine EU-weite Verordnung, die am 25. Mai 2018 in Kraft getreten ist und in der ganzen Europäischen Union, sowie seit dem 20. Juli 2018 auch im Europäischen Wirtschaftsraum, gilt [74]. Es ist eine Vorschrift, die das Datenschutzrecht regeln und dieses innerhalb der EU vereinheitlichen soll. Somit soll zum einen Klarheit gegenüber dem Datenschutz für die Unternehmen innerhalb des Wirtschaftsraumes Europa geschaffen und zum anderen dem Bürger wieder die Hoheit über dessen Daten zurückgeben werden. Dabei wird in der DSGVO speziell der Begriff *personenbezogene Daten* definiert, so dass klar wird, welche Daten genau von dieser Verordnung geschützt werden [75]. Verschiedene Rechte wurden dazu in der Vorschrift definiert, wie zum Beispiel das Recht auf Information und Auskunft, das Recht auf Berichtigung und Löschung oder ein Recht auf Einwilligung zur Datenverarbeitung [76]. Daraus resultierten nun die sogenannten *Cookie-Notices* und *Cookie-Walls*, die vor dem möglichen Zugriff auf eine Website dem Nutzer angezeigt werden, damit dieser auswählen kann (Opt-in), welche Daten von ihm durch die Website gesammelt werden dürfen, bevor die Website dann im vollen Funktionsumfang dem Nutzer freigeschaltet wird. Die europäische Kommission veröffentlichte nach etwas mehr als zwei Jahren einen Pflichtmäßigen Evaluationsbericht zur verabschiedeten Verordnung. Dieser fiel weitestgehend positiv aus, es wurden die meisten Ziele erreicht und die Vereinheitlichung zwischen den EU-Mitgliedsstaaten nahm laut der Kommission zu [77]. Auf das positive Ergebnisse folgte aber auch Kritik, unter anderem, dass die DSGVO zu bürokratisch ist, internationale Fälle zu langsam voranschreiten und allgemein wurde deshalb das Ergebnis anscheinend zu positiv formuliert [78]. Auch schon vor Inkrafttreten der Vorschrift gab es Kritik gegenüber dem großen Spielraum der durch das Gesetz geschaffen wird und die entstehende Bürokratisierung [79]. Die Online Advertising-Branche bereitete sich auf die kommende Verordnung früh vor und es war schnell klar, dass die Regelungen sofortige Auswirkungen auf die Branche haben werden [80]. Doch insgesamt werden viele Themen des Web Trackings in der DSGVO nicht ganzheitlich geklärt. So muss beim Thema Cookies ein Urteil des Europäischen Gerichtshofs aus dem Jahr 2019 Klarheit schaffen. Explizit soll eine weitere Verordnung das Web Tracking besser regeln und eingrenzen: Die ePrivacy-Verordnung. Diese soll den Fokus auf das digitale Geschäft und die Kommunikation setzen und einiges besser definieren. Eigentlich sollte diese Verordnung schon zusammen mit der DSGVO im gleichen Jahr verabschiedet werden, doch gab es bis 2021 noch keine Fortschritte bei diesem Plan [81]. Nachdem Deutschland und

Kroatien mit ihren Vorschlägen gescheitert sind, arbeitet nun Portugal derzeit an einem neuen Vorschlag für die EU [82]. Insgesamt sind also die Meinungen noch immer gespalten, was die DSGVO und dessen Wirkung angeht. Quellen zufolge wurden über 25.000 DSGVO-Fälle 2020 gemeldet, was weitaus mehr als die Jahre davor ist und die deutschen Datenschutzbehörden gaben bekannt dass im Jahr 2020 48,1 Millionen Euro Bußgelder zwecks DSGVO-Verstößen verhängt wurden [83]. Doch eine Studie aus 2021 zeigt nun , dass mehr als 75 % der Tracking-Aktivitäten auf Webseiten schon vor dem Opt-in des Nutzers ausgeführt werden, woraus man schließen kann, dass viele immer noch Probleme haben die Verordnung umzusetzen oder die finanzielle Gefahr nicht begreifen [23].
Im Vergleich zur DSGVO in der EU gibt es auch Fortschritte in anderen Ländern. So ist beim EU-Nachbarn und Mitglied des Europäischen Wirtschaftsraums Norwegen Datenschutz schon lange Gang und Gebe. Seit 1980 ist die Datenschutzbehörde *Datatilsynets* in Norwegen eingerichtet und hat einige Dinge aus der DSGVO mit übernommen [84]. Seit dem werden immer wieder Nachrichten über Strafzahlungen großer Firmen in Norwegen bekannt, welche mit den Datenschutzregelungen im Land zusammenhängen. Der Forendienst *Disqus* muss deshalb 2,5 Millionen Strafe in Norwegen zahlen [85] und der Dating-App *Grindr* drohen Strafgelder in Millionenhöhe [86]. In den Vereinigten Staaten ist der Bundesstaat Kalifornien mit dem *CCPA* aufgefallen. Dieser wirkt wie eine einfache Form der DSGVO, unterscheidet sich aber in der Definition der personenbezogenen Daten und dem Datenschutz für Geräte und Familien. Der CCPA versucht die allgemeinen Datenschutzlücken der Vereinigten Staaten im Bundesstaat zu flicken und es könnte sein, dass in naher Zukunft weitere Bundesstaaten mit einem ähnlichen Gesetz nachziehen werden, eventuell auch landesweit ein solches Gesetz eingeführt wird. Neben den Vereinigten Staaten arbeitet auch Indien mit der *India Personal Data Protection Bill* an einem DSGVO-ähnlichen Datenschutzreglement und Singapur verabschiedete mit dem *Singapore Personal Data Protection Act* einen Zusatz zum schon bestehenden Gesetz aus dem Jahr 2012 [87].

Adblocker als Gegenspieler des Web Trackings

Zuletzt ist auch das Thema *Adblocker* anzusprechen. Auch beim Web Tracking spielen die Adblocker eine wichtige und kritische Rolle. Laut einer Studie aus 2019 nutzen etwa 49 % der Deutschen einen Adblocker, was genau im weltweiten Durchschnitt liegt [88]. Meist sind dies einfache kostenlose Browser-Erweiterungen, welche im Browser installiert werden können. Das Problem ist, zum einen blocken sie Online-Werbung auf Basis von Filter-Listen, wodurch somit der Online Advertising-Branche geschadet wird. Zum anderen werden aber auch bei manchen Adblockern Webanalyse-Dienste wie Google Analytics

geblockt und machen damit Web Tracking unmöglich. Dies geschieht aber nur, wenn es sich eben wie bei Google Analytics um Drittanbieter-Hostings der Web Analytics-Platfform handelt. Bei der Nutzung eines selbst gehosteten Web Analyse-Dienstes wie PIWIK wurde das Web Tracking nicht geblockt. So fand ein Blogger beim Vergleichstest heraus, dass Google Analytics in zwei Tagen 50 % weniger Webseitenaufrufe erkannt hat, als dessen Konkurrent PIWIK. Filterlisten haben dabei die Tracking-URLs von Google Analytics aufgenommen und blocken daher jede Datei, welche von diesen URLs auf der Webseite geladen werden soll [89]. Beim selbst hosten eines solchen Dienstes kann keine Filterliste die URL wissen. Adblocker wie der Stiftung Warentest-Sieger uBlock Origin [90] nutzen dabei eine Vielzahl dieser Filterlisten und sind dabei sehr effektiv Web Tracking abzuhalten. Der Adblocker uBlock Origin ist dazu ein Vorreiter in der Werbeblocker-Branche und bietet weitere Funktionen, wie einen Schutz vor IP-Adressen-Leaking über das in den meisten Browser eingebaute WebRTC-Protokoll an [91] oder einen Blocker gegen das neuartige *CNAME Cloaking*, was eine Möglichkeit ist Third-Party-Tracking über servertechnische Einstellungen als First-Party-Tracking zu verschleiern. Dafür nutzt der Entwickler eine Schnittstelle im Firefox-Browser, weshalb dieses Feature nur für die kostenlose uBlock Origin-Erweiterungen für den Firefox Browser erhältlich ist [92]. Die Entwickler des Brave-Browser sind derweil die ersten, die es geschafft haben einen Blocker für das CNAME Cloaking in ihrem Chromium-basierenden Browser zu entwickeln [93]. Die Adblocker sind somit ein weiterer Mitspieler im Web Tracking-Geschäft und es gilt zu beobachten, was in Rahmen des Web Tracking-Blocking für Fortschritte gemacht werden.

4 Tracking-Arten und ihre technische Umsetzung

4.1 Serverseitiges Tracking

Das *serverseitige Tracking* ist eines der zwei verschiedenen Hauptarten des Trackings. Dazu gibt es noch Mischformen aus beiden Tracking-Arten (hier bezeichnet als „Andere Tracking-Arten“). Als Synonym für das serverseitige Tracking kann hier auch *serverseitige Datensammlung* angewendet werden, denn letztendlich werden beim Tracking Daten über einen Nutzer gesammelt. Bei dieser Art des Trackings werden nur über den ursprünglichen Webserver Daten gesammelt, was mittels Server-Logfiles durchgeführt wird. Diese werden im Anschluss näher erläutert. [1]
Oft wird der Begriff serverseitiges Tracking im Bezug auf den Web Analytics-Dienst *Google Analytics* und dessen Analytics-Server genutzt. Normalerweise werden bei solchen Analytics-Diensten Daten über sogenannte Tags (eine Tracking-Methode des clientsei-

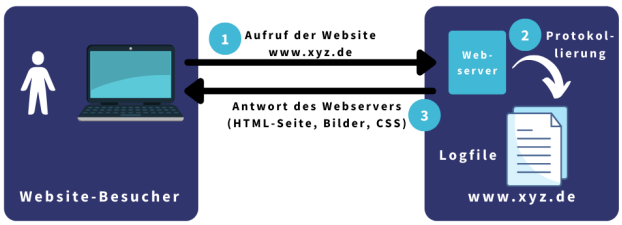

Abbildung 4.1: Funktionsweise der Logfile-Erstellung, auf Basis von [1]

tigen Trackings) gesammelt, welche im späteren Verlauf im Detail erklärt werden. Die gesammelten Daten gehen dann direkt an den Analytics-Server des Anbieters. Mit dem serverseitigen Tracking wird aber der Tag auf dem Webserver gesetzt und die gesammelten Daten gehen dann erst zum Analytics-Server [94]. Diese Art des Trackings lässt sich aber auch ohne clientseitiges Tracking-Methoden umsetzen, was auch in diesem Kapitel erläutert wird. Man sieht hier also, dass die Grenzen der Tracking-Arten immer mehr verschwimmen und Methoden aus anderen Bereichen genutzt werden, um das Maximum für das Tracking herausholen zu können.

4.1.1 Logfiles

Logfiles sind ein zentraler Baustein einer jeden Anwendung oder eines jeden Servers. Sie werden auch *Ereignisprotokolldateien* oder *Log-Dateien* genannt. Im allgemeinen halten sie alle Prozesse fest, die in einer Anwendung im Fokus stehen [95]. Ein Beispiel dafür ist eben die Protokollierung der Ereignisse auf einem Webserver. Dabei helfen diese Log-Dateien meist bei der Wiederherstellung der richtigen Daten nach Abstürzen der Anwendung[95]. Im Falle eines Webserver können sie auch zur Analyse von Clicks bzw. Hits auf einer Website genutzt werden. Logfiles werden normalerweise automatisch erzeugt und beinhalten meist zwei typische Angaben: Das erfasste Ereignis und einen Zeitstempel für das jeweilige Ereignis [95].

Die automatische Erstellung kann je nach Anwendungsgebiet ein wenig abweichen. Im

Abbildung 4.2: Beispiel einer Logfile eines Webservers

Falle des Nutzer-Trackings sind aber nur die Logfiles eines Webservers im weiteren Verlauf interessant. Hierbei läuft die Logfile-Erstellung immer nach einem Schema ab: Ein Website-Besucher gibt die URL einer Website ein und möchte diese besuchen. Dadurch wird ein Aufruf (engl. request) vom Browser des Besuchers an den Webserver der Website gesendet (1), welcher daraufhin mit der gewünschten Webpage antworten und diese zurücksenden möchte. Da eine Website aus vielen verschiedenen Dateien und Bildern besteht, sendet der Browser weitere Anfragen, bis der Webserver mit allen Dateien geantwortet hat und die volle Webpage geladen ist (3). Bei jeder Antwort des Webservers an den Besucher-Browser protokolliert der Webserver Informationen zu den Anfragen und Antworten in einer Logfile (2) (siehe Abbildung 4.1) [1].

Visuell kann eine Log-Datei wie in Abbildung 4.2 aussehen. Letztendlich ist diese einfach nur eine Texdatei, die täglich neu aufgesetzt wird. Jede aufgerufene Datei, was von HTML-Seite, über Cascading-Style-Sheets-(CSS-)Formatdatei, bis hin zu einfachen Bildern und Javascript-Code reicht, wird dabei strukturiert in der Logfile abgespeichert. Ein solcher Aufruf einer Datei wird auch *Hit* bezeichnet [1]. Für jeden Hit werden Informationen, wie unter anderem Datum und Zeit des Aufrufs, IP-Adresse, URL (sozusagen der Dateipfad) der Datei, Daten zum Browser und Betriebssystems des Nutzers, Herkunft des Aufrufs (Referer), Status bzw. Ergebnis und auch erstellte Cookies (siehe Abbildung 4.2)

4.1.2 Trackingmöglichkeiten mit Logfiledaten

Unter den gespeicherten Informationen können einige für das Tracking interessant sein. Allen voran war die mitgespeicherte IP-Adresse in der Vergangenheit wichtig, denn durch sie konnte man Rechner oder Netzwerke eindeutig identifizieren. Der Webserver nutzt dabei diese Adresse, um zu wissen wohin die Dateien gesendet werden sollen. Die IP-Adresse kann aber auch genutzt werden, um eindeutig Personen auf der Website zu ver-

folgen und später bei Logfile-Analysen detaillierte zu analysieren. Mittlerweile ist dies aber aus verschiedenen Gründen nicht mehr in dieser Art möglich. Zum einen gibt es fast keine einzigartigen IP-Adressen mehr. Da die klassischen IPv4-Adressen im Grunde gesehen vier Zahlen zwischen 0 und 255 mit einem Punkt getrennt (Bsp. 192.168.178.02) darstellen wird hier nur eine Länge von 32 Bits (binär) erreicht, was eine Maximalanzahl von 4.294.967.296 IPv4-Adressen ermöglicht [96]. Mit über sieben Milliarden Menschen auf der Erde, welche teilweise mehrere elektronische Geräte besitzen, wurden schnell diese vier Milliarden-IP-Adressen-Grenze erreicht und es wurde als neue Lösung die IPv6-Adresse mit 128 Bits Länge eingeführt. Diese ist jedoch noch nicht so verbreitet. Die Internet Service Provider (ISP), welche die IP-Adressen an die Internetnutzer verteilen, haben sich daher eine Zwischenlösung überlegt. Die ISPs vergeben dabei in bestimmten Zeitintervallen (meist in 24 Stunden oder Sitzungsweise) jedem Endbenutzer eine neue IP-Adresse, wodurch nicht genutzte IP-Adressen wiederverwertet werden und sich die ISPs damit Adressen sparen. Solche IP-Adressen werden *dynamische IP-Adressen* genannt und sind meist im Privatkundenbereich ausschließlich in Benutzung. Firmen und vor allem Webserver von im Internet erreichbaren Webseiten halten statische IP-Adressen, die sich nicht ändern können, inne [97]. Man könnte meinen somit wäre es einfach als Business-to-Business-Unternehmen andere Unternehmen und deren Mitarbeiter zu tracken, doch durch die oft angewendete Funktion eines Firmennetzwerk werden meist keine einzigartige IP-Adressen an alle Mitarbeiter verteilt. Die verteilten IP-Adressen sind dann nur eindeutig innerhalb des Firmennetzwerks, nach außen erscheinen aber alle Anfragen von einer einzigen IP, also die des ganzen Firmennetzwerks [1].

Dazu kommen die schon angesprochenen Probleme bezüglich dem Datenschutz, da die IP-Adresse laut der DSGVO zu den personenbezogenen Daten zählt und deshalb das Tracking der IP für Analysezwecke Einwilligung des Nutzers benötigt. Resultierend aus den oben genannten Argumenten ist heutzutage problematisch die IP-Adresse als einzige Datenquellen zu nutzen, um ein 100%-prozentiges Tracking stellen zu können.

Für kleinere aggregierte Analysen kann man den URL-Eintrag in der Logfile nutzen. Sie zeigt die Herkunft der geladenen Datei an, wodurch man z.B. bestimmen könnte, wie viele Hits auf eine Webpage mit einem speziellen Angebot (Landing Page) erreicht wurden.

Dazu ist die Angabe des User Agent oft eine wichtige Informationen, um User richtig bestimmen zu können. Der User Agent ist dabei lediglich die Information über Betriebssystem, Browser und dessen jeweiligen Versionen, wie auch einige weitere komplementäre Informationen zu den beiden wichtigsten Angaben. Der User Agent kann beispielsweise

so aussehen:

```
Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:47.0)
Gecko/20100101 Firefox/47.0
```

Wichtig sind diese Informationen für den Webserver, um die richtigen Daten an den jeweiligen Browser und das jeweilige Betriebssystem aussenden zu können [1]. Oft wird der User Agent im Falle des passiven Fingerprinting genutzt, was im späteren Verlauf noch weiter erklärt wird. Denn gepaart mit weiteren Informationen kann dieser User Agent genutzt werden, um Nutzer eindeutig und das über mehrere Webseiten hinweg identifizieren zu können. Auch für die Darstellungsoptimierung für mobile Geräte ist der User Agent eine oft genutzte Informationsquelle.
Des Weiteren ist die Information über den Referer auch wertvoll. Dieser besagt, welche Website kurz vor dem Klick auf die eigene Website von einem Besucher besucht wurde [1]. Dabei wird aber immer nur die letzte Website dargestellt, ein tieferer Einblick in die Vergangenheit, wie bei einem Suchverlauf, ist aus Privatsphäregründen nicht möglich. Bedeutsam kann das vor allem bei Analysen der Traffic-Herkunft sein. Für reines User-Tracking ist der Referer aber eher irrelevant.
Zuletzt müssen die Cookies als Information in den Logfiles angesprochen werden. Letztendlich sind Cookies nur kleine Textdateien, die Zeichenketten lokal auf dem Computer eines Besuchers speichern [1]. Es können eindeutige Zeichenfolgen für teilweise unbestimmte Zeit abgespeichert werden, um somit Besucher über mehrere Sitzungen hinweg zu erkennen. Doch das Stichwort ”lokal” zeigt hierbei schon, dass die Setzung eines Cookies auf dem Client, also dem Computer des Besuchers, geschieht, was das Cookie-Tracking zu einem clientseitigen Tracking macht. Es gibt Möglichkeiten, Cookies auch serverseitig zu implementieren, doch wird dies erst im späteren Verlauf nach detaillierter Betrachtung von Cookies verdeutlicht.

Im vergangenen Jahrzehnt wurde das serverseitige Tracking immer unbedeutender, da die Limitierung auf die Server-Informationen als Nachteil angesehen wurde. Logfiles werden nur um weitere Einträge erweitert, wenn ein Besucher neue Unterseiten öffnet und somit mehr Anfragen an den Webserver stellt. Dabei werden Aktionen, wie ein Klick auf einen Button auf der Website oder andere Events, die zwischen dem Aufruf zweier verschiedener Webpages geschehen, gar nicht aufgezeichnet. Nur anhand der Logfile-Daten beispielsweise Änderungen am Content der Webpage durchzuführen macht wenig Sinn, da die Daten dafür nicht wirklich genutzt werden können. Nur die Analyse der getrackten Daten über Logfiles macht das serverseitige Tracking nützlich, jedoch ist das gleiche

auch mit clientseitigen Methoden möglich.
Durch die neueren Datenschutzverordnungen, die nun überall auf der Erde durchgesetzt werden, tendieren mittlerweile zudem immer mehr Tracking-Anbieter und auch Unternehmen, die User tracken möchten, wieder zu einem serverseitigen Tracking, so z.B. Google, wie schon oben mit dem Tag-basierten serverseitigen Tracking für Google Analytics angesprochen.

4.1.3 Bedeutung der Logfile-Daten für das Web Tracking

Die durch die Server-Logfiles getrackten Daten, können so einfach heruntergeladen und lokal für Analysen genutzt werden, um somit den ganzen Kreislauf des Web-Trackings genug zu werden. Das Problem dabei ist aber die lokale Durchführung von Analysen, was nicht gut skalierbar ist. Zumal Logfiles als einfache Textdatei keine wirkliche Formatierung besitzen und man dazu programmiertechnisch die Datei anpassen muss, damit man die Daten wirklich nutzen kann. Daher gilt eher die Anforderung, die getrackten Daten online, zentral und strukturiert für gemeinsame Arbeiten an Analysen aufzubewahren und verfügbar zu machen. Die einfachste Lösung dafür ist die Logfile-Datei mitsamt ihrer Daten in eine Datenbank auf dem Webserver zu speichern, um dadurch dann die Möglichkeit zu bekommen die Daten besser analysieren zu können.
Dies kann zum Beispiel durch serverseitigen PHP-Code durchgeführt werden. Im Beispiel eines Apache-Webserver, die wohl bekannteste Webserver-Software, können die Daten in der Besucher-Logfile (*(*access log)) in eine MySQL-Datenbank des Servers importiert werden. Dazu wird ein PHP-Skript genutzt, dass den Import der Logfile in eine Datenbank übernimmt [98]. Dieses Skript kann man dazu dann über CronJobs auf dem Webserver automatisieren. CronJobs sind hierbei einfach sich wiederholende Aufgaben, die auf unixartigen Betriebssystemen (Linux, Mac OS x) oder Serverumgebungen automatisiert werden können [99]. Somit könnte das Skript so automatisiert werden, dass jeden Tag die neueste Logfile in eine Datenbank importiert wird.
Von hier aus kann nun auf verschiedene Arten der Prozess des Web Trackings fortgeführt werden. Theoretisch können nun in der erstellten MySQL-Datenbank mit SQL-Abfragen Analysen durchgeführt werden, womit das Web Tracking als Prozess beendet wäre. Bei wenigen Daten von kleinen Webseiten ist dies wahrscheinlich sogar eine gute Möglichkeit. Doch bei großen Datenaufkommen ist diese Möglichkeit eventuell nicht skalierbar genug, weshalb man hier mit Pipelines in andere Analysetools skalierbare Anwendungsmöglichkeiten schaffen könnte. Zum Beispiel könnten die Datenbankeinträge in Google BigQuery geladen werden, ein Cloud Data Warehouse von Google, dass dafür gemacht wurde, um große Mengen an Daten effizient zu analysieren. Dabei werden die Daten

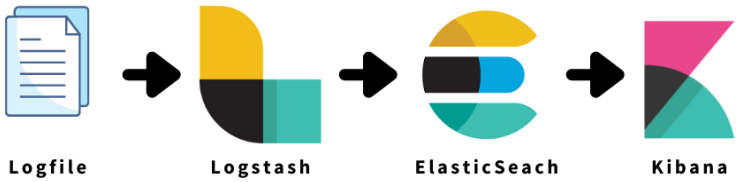

Abbildung 4.3: Architektur des ELK Stacks, auf Basis von [5]

auch mit SQL-Abfragen analysiert [100]. Es kommt hier eben auf das interne Setup des Unternehmens mit ihrer Website und ihren Data Pipelines im Backend an. Es gibt neben den eher einfach gehaltenen Beispielen von oben, ganze Techstacks, die sich mit dem Data Pipelining-Prozess mit Hilfe verschiedener Software beschäftigen. Der bekannteste ist dabei der *ELK Stack*. ELK ist hierbei die Abkürzung für die drei genutzten Produkte *ElasticSearch*, ein Tool zur Speicherung von Logfiles, *LogStash*, eine Software für die Verarbeitung von Daten aus den Logfiles, und *Kibana*, ein Datenvisualisierungstool (siehe Abbildung 4.3). Diese Architektur der Pipeline kann auf verschiedene Weise mit weiteren komplementären Tools erweitert und optimiert werden, was aber über den Rahmen des Themas hinausschreitet [5]. Auch die Nutzung von Amazon Web Services Tools zum Pipelining von Logfile-Daten in Analysetools sind möglich [101].

4.2 Clientseitiges Tracking

Das *clientseitige Tracking* ist nun die komplementäre Tracking-Art zum serverseitigen Tracking. Der große Nachteil des serverseitigen Trackings, nur die Besucher aus Sicht des Servers zu betrachten und wichtige Schritte beim Aufenthalt auf der Website nicht zu tracken, hat das clientseitige Tracking zur wichtigsten Art des Trackings gemacht. Das clientseitige Tracking beschreibt daher, dass Tracking nicht auf dem Server durchgeführt wird sondern auf dem Client, also dem Browser des Besuchers. Jede Benutzertätigkeit wird dann über den Browser protokolliert und gesammelt. Diese Tätigkeiten können daher neben Mausklicks, -position oder -bewegungen auch Tastatureingaben, Fenstergröße oder Auflösung des Bildschirms beinhalten und selbst diese Liste ist noch nicht erschöpft [1]. Dies sind alles Daten, die beim serverseitigen Tracking nicht gesammelt werden können, aber im Gegenteil dazu sind alle Daten aus den Log-Dateien beim serverseitigen Tracking (URL, Referer, User Agent) auch beim clientseitigen Tracking vorhanden. Man bekommt also so ziemlich alle Daten beim clientseitigen Tracking, wohingegen man beim serverseitigen Tracking limitiert ist.
Warum das funktioniert liegt vor allem an den Gegebenheiten eines Browsers. Sobald

eine Website in einem Browser geladen wurde, kann die Seite mit Hilfe von JavaScript die Browsereinstellungen und ausgeführte Aktionen beobachten und aufzeichnen [1].

4.2.1 Page Tagging

Zwar sind die Informationen über den Besucher via clientseitigem Tracking relativ einfach zu erheben, stellt sich die Informationsweitergabe an den Webserver oder das Unternehmen, das die Website betreibt, als Herausforderung dar. Um diese Aufgabe durchführen zu können, hat man das *Page Tagging* erfunden. Page Tagging beschreibt den Einbau eines sogenannten Tags auf der Website, mit welchem die gesammelten Daten zum Website-Betreiber übertragen werden können. Ein Tag ist dabei einfach nur ein Schlüsselwort, das zur Beschreibung eines Elements auf einer Webpage und aller seiner Attribute dient [102]. Die Erweiterung des Begriffs zu Page Tag beschreibt daher dann die Kennzeichnung einer ganzen Website und nicht einem Element der Website. Meist wird dann im gleichen Zug der Begriff *Pixel* auch erwähnt und teilweise als Synonym mit einem Tag gehandelt. Doch dem ist nicht so. Die Kennzeichnung eines Element kann auf verschiedene Weisen geschehen, so ist auch ein Anchor-Tag in HTML auch per Definition ein Tag. Der Pixel ist daneben auch ein Tag, aber kein Synonym, da wie gezeigt, ein Tag nicht immer ein Pixel sein muss. Pixel sind schon seit den 1990er-Jahren durch die sogenannten Zählpixel bekannt [103]. Letztendlich sind Pixel, wie der Name schon preis gibt, 1x1-Pixel große Bilder, die somit in der Praxis unsichtbar für den Nutzer einer Website sind, gepaart mit JavaScript-Code für die Tracking-Logik [1]. Die Zählpixel waren damals noch sichtbar und wurden dazu genutzt, um jeden Seitenaufruf zu dokumentieren und somit erste einfache Analysen für die Website zu erstellen. Auch wurden die Zählpixel *Web Bugs* (Web-Wanzen) oder auch *Web Beacon* (Web Signalfeuer) als Synonyme eingebürgert [1], wodurch sich aus dem letzteren wahrscheinlich auch die Begrifflichkeit des "Feuerns eines Pixels" (engl. fire a pixel) etabliert hat, wenn man von der Aktivierung eines Pixels sprechen möchte.
Die Funktionsweise des Page Tagging als Prozess ist dabei auch eher unkompliziert. Ein Besucher ruft eine Website auf, wodurch Anfragen beim Webserver entstehen (1). Dieser Webserver antwortet mit den jeweiligen Dateien, die der Besucher benötigt, darunter auch die HTML-Datei mit dem eingebauten Page Tag in Form eines JavaScript-Codes(2). Der JavaScript-Code wird meistens in den *<head>*-Bereich im HTML-Dokument eingefügt, kann aber auch am Ende des *<body>*-Elements auftauchen. Sobald nun die gesendeten Dateien des Webservers im Browser geladen werden, wird der JavaScript-Code ausgeführt und sammelt nun die gewünschten Daten über den Besucher und lädt am Ende den Pixel in Form eines *img*-Element (*iframe*-Elemente sind auch möglich) über einen

Drittserver (auch Analytics-Server genannt) auf die geladene Seite. Durch das versetzte Laden können nun die gesammelten Daten an den Drittserver gesendet werden (3). Dabei werden die Informationen in der URL des Pixel-Bildes als Parameter hinzugefügt, so dass diese vom Drittserver erkannt und abgespeichert werden können.

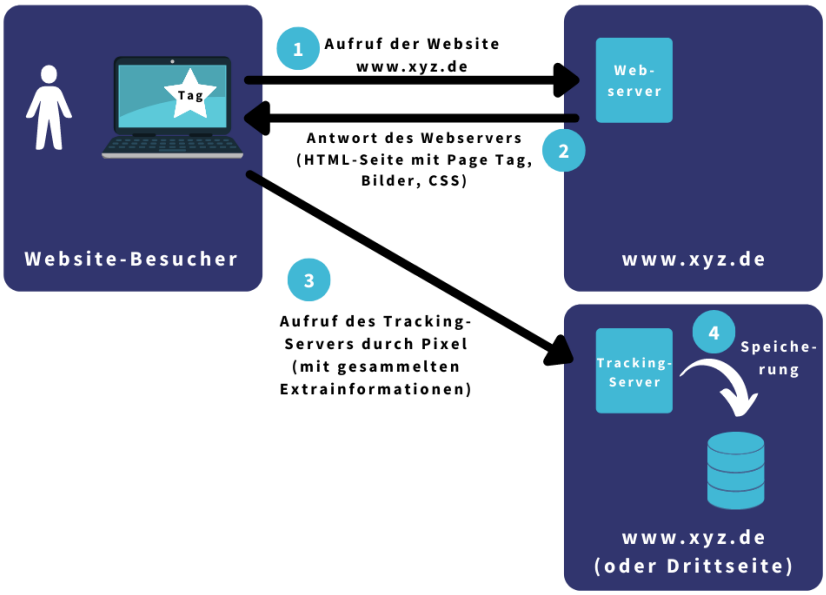

Abbildung 4.4: Funktionsweise des Page Tagging, auf Basis von [1]

Ein Beispielcode für Page Tagging ist in Abbildung 4.4 zu sehen. Hier werden erst verschiedenen Informationen des Clients getrackt und in Variablen abgespeichert. Darunter der Browser und die Browser-Version abgeleitet aus dem User Agent, die URL der Website und eine erstellte SessionID welches eine einzigartige Folge an Zahlen und Buchstaben ist. Daraufhin wird dann ein *img*-Element erstellt, welchem eine Quelle für das zu ladenden Bild hinzugefügt wurde. Dabei wurde die URL für das Bild mit Parametern, die aus den zuvor gesammelten Daten des Clients bestehen, erweitert. Dieses *img*-Element wird dann noch die Höhe und Breite von 1 zugewiesen, bevor es dann als Knoten hinter dem letzten Knoten beim angegebenen Element im Document Object Model-(DOM-)Baum angefügt wird.

```
<div id="container">
    <h1>Test-Website</h1>
    <p>Das ist eine Test-Website für das Page Tagging.</p>
</div>
<script>
    ! function () {
        // track data
        // browser and browser version
        var browser_details = function() {
            var ua = navigator.userAgent;
            var M = ua.match(/(opera|chrome|safari|firefox|msie|
                trident(?=\/))\/?\s*(\d+)/i);
            M = M[2] ? [M[1], M[2]] : [navigator.appName,
                navigator.appVersion, "-?"];
            return M.join("_");
        };
        // website url
        var url = window.location.href;
        // unique session id
        var ID = function() {
            return "_" + Math.random().toString(36).substr(2, 9);
        };
        // create pixel
        var img = document.createElement('img');
        img.src = "https://www.tracking-server-test.de/pt.png?sid=" +
             ID() + "&url=" + url + "?browser_version=" +
            browser_details();
        img.height = "1";
        img.width = "1";
        document.getElementById('container').appendChild(img);
    }();
</script>
```

Quellcode 1: Ausschnitt eines Beispielcodes zur Funktionsweise des Page Tagging

Die letztendliche URL die zum eigentlichen Tracking-Server (Drittserver) führen soll, würde in diesem Beispiel auf einem lokalen Server laufend beispielsweise so aussehen:

`https://www.tracking-server-test.de/pt.png?sid=_vspwdnfwi&url=http://127.0.0.1:5500/pagetagging.html?browser_version=Chrome_91`

Alle getrackten Informationen wurden in der URL des Pixels eingefügt. Auf dem Tracking-Server muss nun ein Server-Skript dazu noch aktiv sein, dass die Parameter ausliest und in einer Datenbank speichert, so dass die Daten dann wieder zu Analysezwecken genutzt werden können.

Nachteil an den Page Tags ist, dass wenn man darüber Daten der Besucher auf der ganzen Website tracken möchte, auch auf jeder Unterseite der gleiche Quellcode eingefügt werden muss. Bei Privatprojekten oder Kleinunternehmen mag das vielleicht kein Problem sein, doch kann das für große Unternehmen ein zu großer Aufwand sein. Dagegen helfen mittlerweile Website-Templates oder Content Management Systeme, bei welchem man den Code-Schnipsel hinterlegen kann, und dies somit auf allen Seiten übernommen wird. Dazu gibt es auch Tag-Management-Systeme. Wenn man vor allem Tags von verschiedenen Anbieter von beispielsweise Ad-Tech-Anbietern auf der eigenen Website einfügen möchte, kann die Anzahl an Tags, die auf allen Unterseiten implementiert werden müssen, sich stark erhöhen. Tag-Management-Systeme, können dabei helfen die Tags auf die richtigen Unterseiten zu implementieren und sie im allgemeinen besser verwalten zu können. Technisch gesehen wird von dem Tag-Management-System, wie z.B. Google Tag Manager, ein einzelner Page Tag auf der ganzen Website integriert, über den nun andere Tags nach Belieben zu unterschiedlichen Unterseiten hinzugefügt werden können [1].

4.2.2 Cookie-Tracking

Eine weitere clientseitige Tracking-Art ist das *Cookie-Tracking*. Theoretisch handelt es sich bei Cookies nur um eine Speicherform von einigen wenigen Daten im Browser des Clients, doch die Nennung des Cookie-Trackings als eigene Tracking-Form liegt an der variierenden Aufteilung der verschiedenen Tracking-Arten in der gängigen Literatur. Während diese Abhandlung sich auf die Unterschiede des serverseitigen und clientseitigen Tracking fokussiert, können die verschiedenen Tracking-Arten auch in ihre Beständigkeit bei der Speicherung aufgeteilt werden. So würde es die Session-basierten-Tracking-Methoden geben (hier nicht einzeln behandelt, da Session-Tracking meist mit Cookies umgesetzt wird), Speicher-basierte-Tracking-Methoden (u.a. Cookies), Cache- bzw. Zwischenspeicher-basierte-Tracking-Arten und auch Fingerprinting-Methoden als einzelne Kategorien nach der alternativen Einteilung geben.
Doch nachdem Cookies einmal definiert werden, wird es deutlich, warum diese zum clientseitigen Tracking zählen können. Cookies sind kleine Textdateien (maximal 4 Kilobyte groß), die in einem Browser durch einen Webserver platziert werden. Sobald ein Nutzer eine Website zum ersten Mal besucht wird ein Cookie (meist gefüllt mit einer einzigartigen ID) im Browser des Nutzers gesetzt. Die Website bzw. der zugehörige Webserver kann nun jedes mal wenn der Nutzer wieder die Website besucht diesen Cookie auslesen, solange dieser eben nicht gelöscht worden ist [21]. Allgemein gilt das Cookie-Tracking als sehr schnell und gut in der Praxis anwendbar, da der Nutzer keine bestimmten Aktionen

machen muss, damit das Cookie gesetzt wird und auch so ist das setzen des Cookies völlig unsichtbar für den Endnutzer [21]. Doch Nachteile ergeben sich vor allem aus den jüngsten Veränderung in der Cookie-Tracking-Debatte. Nutzer werden immer aufmerksamer, wenn es um Cookies geht, DSGVO & Co. machen das Setzen von Cookies durch Ablehnung des Trackings durch den Nutzer schwieriger und Browser sperren teilweise Cookies schon komplett (Safari). Cookies funktionieren nur noch in einem ganz bestimmten Fall, und zwar wenn der Nutzer das Setzen eines Cookies zulässt, Cookies nicht aus dem Cache des Browsers gelöscht werden und dazu der Nutzer auch immer den gleichen Browser beim Besuch der Website nutzt [21]. Denn jeder Browser hat einen eigenen "Cookie-Ablageort" und die Browser greifen nicht auf diese von anderen Browsern zu.
Da das Cookie auf dem Client (Browser) gesetzt wird und von dort aus auch ausgelesen werden könnte, kann man demnach davon ausgehen, dass das Cookie-Tracking teil der clientseitigen Tracking-Arten sein kann, obwohl man zum einen auch serverseitig die Cookies lesen und setzen könnte, wie man schon bei den Logfiles gesehen hat und später im Bezug zu den *HttpOnly*-Cookies erkennen kann.
Insgesamt gibt es dabei zwei Arten von Cookies. Einmal gibt es die *Session-Cookies*, welches direkt nach dem Schließen der Website ablaufen und im Browser gelöscht werden [21]. Technisch setzt man einen Cookie einfach ohne ein Ablaufdatum, um ein Session-Cookie zu erstellen, doch die beispielhafte Anwendung von Cookie-Tracking wird im Folgenden noch erläutert. Als zweite Art gibt es die *Persistenten Cookies*. Diese werden mit einem Ablaufdatum gesetzt und laufen eben dann ab, wenn das Ablaufdatum erreicht wurde [104].
Dazu gibt es zwei Arten wie man Cookies setzen und 2 Arten wie man Cookies lesen kann. Zum einen kann ein Cookie über einen einfachen Application Programming Interface-(API-)Call via Javascript gesetzt werden. Cookies werden mit Hilfe einer Web API durch den Befehl *document.cookie* erstellt [105]. Dabei kann man den Befehl einer ganzen Funktion hinzufügen, welche die benötigten Angaben wie den Namen des Cookies, den Wert des Cookies mit dem dazu berechneten Ablaufdatum zusammenfügt, wie im Quellcode 2 zu sehen. Der Code würde so nun automatisch einen Cookie mit dem Namen "Testcookie" und mit dem Wert "Testwert" erstellen, der nach einem Tag ablaufen würde. Dies entspricht einem persistenten Cookie, da das Ablaufdatum gesetzt wurde ("expire ="). Wäre kein Ablaufdatum bei der Erstellung des Cookies involviert würde ein Session-Cookie das Ergebnis der Funktion sein. Wie schon angesprochen würde man zudem in der Praxis als Wert für einen Cookie eher eine einzigartige ID einfügen oder anderweitig aussagekräftigere Daten.

```
<script>
    ! function setCookie(cname, cvalue, exp) {
        const d = new Date();
        d.setTime(d.getTime() + (exp * 24 * 60 * 60 * 1000));
        let expires = "expires=" + d.toUTCString();
        document.cookie = cname + "=" + cvalue + ";" + expires + ";path=/
            ";
    }("Testcookie", "Testwert", 1); // 1 = 1 day expiration
</script>
```

Quellcode 2: Ausschnitt eines Beispielcodes zur Setzung eines Cookies via API-Call, auf Basis von [6]

Die andere Art einen Cookie zu setzen ist die Nutzung des *Set-Cookie*-Header in einer Hypertext Transfer Protocol-(HTTP-)Antwort auf eine Anfrage.

```
Set-Cookie: <cookie-name>=<cookie-value>
```

Hierbei wird dem HTTP-Response einfach der obenstehende Term serverseitig hinzugefügt, so dass ein Cookie zum User Agent des Clients hinzugefügt wird. Der Server kann dann diesen Cookie über die Logfiles, wie schon beim serverseitigen Tracking betrachtet, auslesen [106]. Der serverseitigen Setzung eines Cookies kann dabei noch mit dem *HttpOnly*-Parameter erweitert werden, um eine Art "serverseitiges Tracking" aufzusetzen, wie schon bei den Logfiles angesprochen. *HttpOnly* ist eigentlich ein Schutzmechanismus, der verhindern soll, dass ein clientseitiges Skript auf ein Cookie zugreifen und diesen verändern kann. Durch das Setzen des Parameters kann nur noch der Webserver selbst auf den Cookie zugreifen und verändern. Somit entsteht in etwa ein serverseitiges Tracking, was zum Beispiel beim Session-Tracking angewendet werden kann [107]. Da aber dieser Cookie letztendlich immer noch auf dem Client gespeichert wird, kann aber eher davon abgesehen werden, Cookie-Tracking auch zu den serverseitigen Tracking-Arten zu zählen.

Zu den zwei Arten, wie Cookie gelesen werden können, gehören einmal die Anfrage mit einem API-Call oder die Auslesung des Cookie-Headers, die bei jeder Anfrage der setzenden Domain mitgeführt wird und man diesen dann in der Logfile betrachten kann. Die zweite Art ist somit relativ simpel. Bei der ersten benötigt es zum einen eine Funktion, die den gewünschten Cookie ausliest und dazu eine Möglichkeit, wie der Wert zum Webserver oder einen Drittserver gelangen kann. Um einen Cookie und dessen gespeicherten Wert via API-Call auslesen zu können, muss eine Funktion wie in Quellcode 3 geschrieben werden. Hier wird erst der zu suchende Name des Cookies in einer Variable abgespeichert, bevor dann alle Cookies in einem Array gespeichert werden, welcher dann

nach dem Cookienamen durchsucht wird. Falls das Cookie gefunden wurde, wird er in diesem Fall einfach als ”Alert” im Browser ausgegeben. Wenn nun vorher ein Cookie mit der vorherigen Funktion aus Quellcode 2 gesetzt wurde, dann würde die Funktion aus Quellcode 3 als Ergebnis den Wert ”Testwert” als Alert ausgeben. Hier müsste dann nur noch eine Logik für das Weiterführen des Cookiewerts zu einem Server eingebaut werden. Zum Beispiel könnte man hier eine Page Tagging-Logik einfügen, die ein *img*-Element erstellt und in der URL des darzustellenden Bildes den Cookiewert als Parameter an einen Server weitergibt.

```
<script>
    ! function getCookie(cname) {
        let name = cname + "="; // searched cookie
        let decodedCookie = decodeURIComponent(document.cookie); //
            decode cookies (special chars e.g. $)
        let ca = decodedCookie.split(';'); // split all set cookies
        // search for cookie with cname and alert value
        for (let i = 0; i < ca.length; i++) {
            let c = ca[i];
            while (c.charAt(0) == ' ') {
                c = c.substring(1);
            }
            if (c.indexOf(name) == 0) {
                return alert(c.substring(name.length, c.length));
            }
        }
        return "";
    }("Testcookie");
</script>
```

Quellcode 3: Ausschnitt eines Beispielcodes zum Lesen eines Cookies, auf Basis von [6]

Neben der direkt durch eine Web API in JavaScript verfügbare Möglichkeit Cookies zu setzen, gibt es auch eine sehr bekannte Cookie-API namens *js-cookie*, welche auch auf vielen Seiten anstelle von *document.cookie* genutzt wird [108].

Eine häufig mit Cookies umgesetzte Tracking-Art ist das *Session-Tracking*. Dabei geht es darum den Nutzer einer Website während seinem Aufenthalt (Sitzung) eine eigene ID zuzuweisen, über die alle Aktionen des Nutzers aufgezeichnet werden können. Somit kann durch das Abspeichern von einem virtuellen Zustand ein besseres Nutzererlebnis geschaffen werden. Dies kann über einige wenige Techniken umgesetzt werden. Allen voran kann man mit Hilfe von Cookies eine SessionID abspeichern, die dann mit Nutzerdaten zugehörig zu der SessionID, welche meist in einer Hashmap gespeichert werden,

verglichen werden können. Das Session-Tracking kann aber auch alternativ mit anderen Lösungen umgesetzt werden, wie dem Nutzen versteckter Felder im HTML-Code (Hidden Form Field), dem hinzufügen von Parametern bei einer URL (URL Rewriting) oder bei Webcontainern und Servlets die Nutzung der HTTP Session via Java [109].

4.2.3 Web Storage

Da Cookies nun immer schwieriger einsetzbar sind (rechtlich, wie auch technisch in Browsern wegen Blockierung) sind in den letzten Jahren andere Speicherarten aufgestiegen, die alternativ in manchen Situationen genutzt werden können. Eine dieser Arten ist der Web Storage. Diese API stellt Mechanismen zur Verfügung, durch welche Schlüssel-Werte-Paare in einem Browser abgespeichert werden können [110]. Zwar sind die Web Storage-Mechanismen trotzdem von rechtlichen Einschränkungen nicht geschützt, da sie trotzdem Daten speichern bzw. verarbeiten, doch sind sie sicherer, können mehr Daten abspeichern (bis zu 5 Megabyte (MB)) und behindern die Performanz der Website dabei nicht [7].
Die Web Storage API führt zwei Mechanismen, die die Speicherung von Daten zulassen. Auf der einen Seite gibt es den *localStorage*. Der localStorage hält dabei die *same-origin-policy* ein, welches ein Schutzmechanismus ist, der Einschränken soll, wie ein Dokument, ein Skript oder eben einfach Daten von einem Ursprung von anderen Skripten anderen Ursprungs genutzt werden können [111]. Somit kann man potenziell gefährlichen Skripten, welche von einer anderen Quelle aus geladen wurden, die Möglichkeit nehmen auf Daten innerhalb der eigentlichen Website zuzugreifen. Daten werden im localStorage als Objekt in Form von Schlüssel-Werte-Paare abgespeichert und sind dabei permanent verfügbar. Das heißt den Browsertab oder den ganzen Browser schließen hat keine Auswirkung auf den localStorage. Sobald aber die Website oder der Nutzer den localStorage-Eintrag löscht sind die Daten verschwunden [21]. Auf der Website-Seite geht das über JavaScript, auf der Nutzer-Seite über das Löschen des Browser-Caches [110]. Der 5 MB Speicherplatz ist hier im Vergleich zur Cookie-Größe ein großer Vorteil und das vor allem eher für das Programmieren von Web-Anwendungen, abseits vom Nutzer-Tracking. Dazu kann der Inhalt des localStorage auch über mehrere Browsertabs geteilt werden [21]. Die Abspeicherung von Daten im localStorage ist zudem auch keine Schwierigkeit. Meist prüft man anfangs, ob der genutzte Browser den Web Storage unterstützt, um dann bei Unterstützung mit dem Speicherbefehl den localStorage zu nutzen. Auch die Anfrage für ein Wertepaar kann mit einem Befehl durchgeführt werden. Quellcode 4 zeigt erst den Test auf Web Storage-Unterstützung, um dann erst ein Schlüssel-Werte-Paar abzuspeichern und diesen daraufhin im (nicht dargestellten) HTML-Dokument auszugeben.

```
<script>
    // Check browser support
    if (typeof(Storage) !== "undefined") {
        // Store
        localStorage.setItem("schluessel", "wert");
        // Retrieve
        document.getElementById("container").innerHTML =
            localStorage.getItem("schluessel");
    } else {
        document.getElementById("container").innerHTML = "Sorry, your
            browser does not support Web Storage...";
    }
</script>
```

Quellcode 4: Ausschnitt eines Beispielcodes zur Nutzung des localStorage, auf Basis von [7]

Auf der anderen Seite gibt es dann noch den *sessionStorage*. Dieser Web Storage Mechanismus funktioniert ähnlich wie der localStorage doch existiert nur pro Sitzung. Das bedeutet, dass gespeicherte Daten im sessionStorage nur solange da bleiben, bis der Browsertab oder der ganze Browser geschlossen werden [110]. Auch hier können bis zu 5 MB gespeichert werden und die same-origin-policy wird auch eingehalten [21]. Wie in Quellcode 5 zu sehen, ähneln sich die Implementierung stark, einzig und allein das Objekt wird von localStorage zu sessionStorage geändert.

```
<script>
    // Check browser support
    if (typeof(Storage) !== "undefined") {
        // Store
        sessionStorage.setItem("schluessel", "wert");
        // Retrieve
        document.getElementById("container").innerHTML =
            sessionStorage.getItem("schluessel");
    } else {
        document.getElementById("container").innerHTML = "Sorry, your
            browser does not support Web Storage...";
    }
</script>
```

Quellcode 5: Ausschnitt eines Beispielcodes zur Nutzung des sessionStorage, auf Basis von [7]

Bei beiden Web Storage Mechanismen müssten jetzt die gespeicherten Daten über das Page Tagging-Verfahren oder andere Wege zur Datenübergabe (z.B. POST-Befehl via

HTTP und *Asynchronous JavaScript And XML* (AJAX)) an den Zielserver gesendet werden.

4.2.4 IndexedDB

Ähnlich zum Web Storage wurde eine Möglichkeit benötigt noch mehr als 5 MB-Daten in einem Element abzuspeichern. Anfangs wurde dafür die *Web SQL Database* genutzt, welches eine auf SQLite basierenden relationale Datenbank ist, in welcher man lokal, wie beim Web Storage, Daten speichern kann. Die Entwicklung dessen wurde aber zwecks Veröffentlichung der *IndexedDB* eingestellt und diese hat nun den Platz für die Web SQL Database eingenommen. Die IndexedDB folgt wie der Web Storage der same-origin-policy, kann aber dazu mehr Daten abspeichern [112].Per Definition ist die IndexedDB eine objektorientierte Datenbank, die auf einem Transaktion-Datenbankmodell aufbaut, das heißt alles was in der IndexedDB gemacht wird, steht immer im Kontext einer Transaktion. Dazu kommt noch der Fakt, dass eine IndexedDB immer Anfragen verwendet. Eine weitere Eigenschaft, die sogar sehr zentral ist, beschreibt, dass IndexedDB meist asynchron funktioniert, was sie zu einem sehr fortgeschrittenen Objekt in der JavaScript-Programmierung macht [112].
Die IndexedDB unterstützt ein Grundschema mit dem auf die Datenbank zugegriffen und Änderungen betrieben werden können. Dieses sieht wie folgt aus:

1. Datenbank öffnen und eine Transaktion initiieren
2. Objektspeicher erzeugen
3. Anfordern der Ausführung einer Datenbankoperation (z.B. Hinzufügen oder Auslesen von Daten)
4. Auf richtige DOM-Ereignis als Antwort warten
5. Ergebnisse verarbeiten (z.B. bei Auslesen von Daten, diese dann an Drittserver senden) [113]

Beim Grundschema können jedoch die Initiierung einer Transaktion und das Erzeugen eines Objektspeichers auch vertauscht werden, denn eine Transaktion wird nur für die Ausführung von Datenbankoperationen benötigt. Die IndexedDB ist sehr mächtig und kann durch ihre hohen Speicherkapazitäten für viele Anwendungsmöglichkeiten genutzt werden. Zugleich steigt auch der Aufwand eine IndexedDB-Datenbank aufzusetzen und

mit Daten zu füllen. In den Quellcodes 6 bis 9 wurde daher der Quellcode so weit wie möglich komprimiert, so dass nur das nötigste beispielhaft gezeigt werden kann.
Quellcode 6 zeigt die eigentliche Initialisierung und somit auch den Hauptteil der IndexedDB-Programmierung. Erst wird geprüft, ob die IndexedDB-Datenbank unterstützt wird. Dann wird die eigentliche Datenbank mit dem Namen "Preferences" geöffnet und die Version 1 festgelegt (IndexedDB kann durch Veränderung der Versionsnummer abgeändert werden). Dadurch dass jeglicher Befehl wie als eine Anfrage im IndexedDB-Kontext zu behandeln ist, ist das Ergebnis dieses Befehls einer von drei Ereignissen: *success*, *error* oder *upgradeneeded*. Die ersten beiden sind selbst erklärend, doch der dritte ist besonders. *Upgradeneeded* erscheint als Ergebnis, wenn man eine Datenbank öffnen möchte, die es schon im Browser gibt und dabei eine kleinere Versionsnummer besitzt. Dazu wird dieses Ergebnis auch eintreffen, wenn eine IndexedDB-Datenbank zum ersten Mal eröffnet wird. Deshalb werden im *Upgradeneeded*-Case erst der *Object Store* erstellt. Dies ist im Grunde gesehen eine Tabelle, so wie es im relationalen Datenbankschema üblich wäre. Dazu wird noch ein Index auf eine einzigartige ID gelegt für Suchzwecke.

```
! function() {
    // check for support
    if (!window.indexedDB) {
        alert("IndexedDB wird nicht unterstützt!");
    }
    // open indexedDB database
    const request = indexedDB.open('Preferences', 1);
    // handle error case
    request.onerror = (event) => {
        console.error(`Database error: ${event.target.errorCode}`);
    };
    // handle successful open case (after first time open)
    request.onsuccess = (event) => {
        const db = event.target.result;

        insertPreference(db, getBrowser_SID());
    };
    // handle when new version > old version
    // also handles first time setup
    request.onupgradeneeded = (event) => {
        let db = event.target.result;
        // create the browser preference object store
        // with auto-increment id
        let store = db.createObjectStore('Browser_Pref', {
            autoIncrement: true
        });
```

```
        // create an index on the sessionid property
        let index = store.createIndex('sessionid', 'sessionid', {
            unique: true
        });
    };

}();
```

Quellcode 6: Beispielcode zur Initialisierung einer IndexedDB-Datenbank, auf Basis von [8]

Sobald der *Upgradeneeded*-Case durchgeführt wurde und den ersten Object Store erstellt hat, wird das Success-Ereignis eingeleitet und es wird eine Funktion zur Erstellung eines Datenbank-Eintrags ausgeführt, welche in Quellcode 7 zu erkennen ist. Zuerst wird hier eine Transaktion initiiert, so wie es vom Grundschema bekannt ist. Daraufhin wird der zu befüllende Object Store ausgewählt. Dann werden mit *.put(data)* die gewünschten Daten in den Object Store eingetragen. Dieser Befehl kann dieses mal zwei Status ausgeben, *success* oder *error*. Während im *error*-Case abermals nur der Error-Code ausgegeben wird, wird im *success*-Case einfach die Transaktion als Event in der Konsole des Browsers dargestellt. Zum Schluss muss nach Vollendung der Transaktion die Verbindung zur Datenbank geschlossen werden.

```
function insertPreference(db, data) {
    // create a new transaction
    const txn = db.transaction('Browser_Pref', 'readwrite');
    // get the browser preference object store
    const store = txn.objectStore('Browser_Pref');
    //put data into store
    let query = store.put(data);

    // handle success case
    query.onsuccess = function(event) {
        console.log(event);
    };
    // handle the error case
    query.onerror = function(event) {
        console.log(event.target.errorCode);
    };
    // close the database once the
    // transaction completes
    txn.oncomplete = function() {
        db.close();
    };
```

```
}
```

Quellcode 7: Beispielcodes für das Hinzufügen eines Eintrags in eine IndexedDB-Datenbank, auf Basis von [8]

Im Quellcode 6 konnte man sehen, dass beim Ausführen der *insertPreference*-Funktion, welches die Funktion für das Hinzufügen eines Eintrags in eine IndexedDB-Datenbank darstellt, sehen, dass für den *data*-Parameter eine weitere Funktion aufgerufen wird. In diesem Fall wurde diese Funktion so implementiert, dass sie den Browser und dessen Version und eine einzigartige SessionID, wie aus dem Page Tagging-Beispiel in Quellcode 1, speichert, aber dieses mal als Objekt ausgibt. Dieses Objekt stellt dann die Wertepaare dar, die in den Object Store der IndexedDB-Datenbank gespeichert werden.

```
function getBrowser_SID() {
    // get browser details
    var browser_details = function() {
        var ua = navigator.userAgent;
        var M = ua.match(/(opera|chrome|safari|firefox|msie|trident(?=\/)
            )\/?\s*(\d+)/i);
        M = M[2] ? [M[1], M[2]] : [navigator.appName,
            navigator.appVersion, "-?"];
        return M.join("_");
    };

    //  get unique session id
    var SID = function() {
        return "_" + Math.random().toString(36).substr(2, 9);
    };
    // return object with data
    return {
        sessionid: SID(),
        browser_preference: browser_details()
    };
}
```

Quellcode 8: Beispielcodes für das Generieren von Daten für einen IndexedDB-Datenbankeintrag

Um nun einen Wert aus der IndexedDB-Datenbank auszulesen muss die *insertPreference*-Funktion aus den *success*-Case bei Öffnung der Datenbank mit einer Funktion ausgetauscht werden, die die gewünschten Daten ausliest und anderweitig verarbeitet. Dazu kann *insertPreference(db, getBrowser_SID());* mit dem Befehl *getBrowserpref(db, 1);* ausgetauscht werden, was die Funktion aus Quellcode 9 aufruft. Letztendlich läuft

diese Transaktion genau gleich ab, nur statt eines *put()*-Befehl wird ein *get()*-Befehl aus der IndexedDB-API genutzt. Der Parameter für diesen Befehl beschreibt einfach nur die Position des auszulesenden Wertes in der IndexedDB-Datenbank, ähnlich wie der Index eines Wertes in einem Array. Beim *success*-Case wird nun in diesem Beispiel der Eintrag an diese Position im Object Store einfach in der Konsole des Browsers ausgegeben, doch eigentlich sollte nun hier eine Logik implementiert werden, die die Daten zu einem Server sendet, zum Beispiel über das Page Tagging-Verfahren.

```
function getBrowserpref(db, id) {
    // create a new transaction, readonly mode
    const txn = db.transaction('Browser_Pref', 'readonly');
    // get the browser preference object store
    const store = txn.objectStore('Browser_Pref');
    // get object in store by id
    let query = store.get(id);
    // handle success case
    // here just console log
    query.onsuccess = (event) => {
        if (!event.target.result) {
            console.log(`Preference with ${id} not found`);
            // insert a logic for passing the value
        } else {
            console.table(event.target.result);
        }
    };
    // handle error case
    query.onerror = (event) => {
        console.log(event.target.errorCode);
    };
    // close connection
    txn.oncomplete = function() {
        db.close();
    };
};
```

Quellcode 9: Beispielcodes für das Hinzufügen eines Eintrags in eine IndexedDB-Datenbank, auf Basis von [8]

Diese ausführliche Abhandlung der IndexedDB ist vor allem wegen der enormen Wichtigkeit dieser in der Zukunft nötig. Eine Studie hat herausgefunden, dass 2018 bei den Top 10.000 Alexa-Seiten mehr 11 % dieser Seiten die IndexedDB-Funktionalität nutzen, was in Anbetracht der schwierigen Situation der Cookies noch steigen könnte, auch wenn jetzt die IndexedDB noch wenig für das Tracking genutzt wird [114].

4.2.5 Fingerprinting

Die wohl gerade bekannteste Tracking-Art neben dem Cookie-Tracking ist das *Fingerprinting*. Synonym wie *Digital Fingerprinting* [115] oder *Browser Fingerprinting* [116] werden auch angewendet. Ein Fingerabdruck (engl. Fingerprint) ist eigentlich der Abdruck der Innenflächen der Finger, welcher zur Feststellung der Identität einer Person genutzt werde kann [117]. Dabei ist es möglich eine Person durch ihren Fingerabdruck eindeutig zu bestimmen, ohne dass eine andere Personen den gleichen Abdruck besitzt. Diese Analogie wurde in die digitale Welt in den Nutzer-Tracking-Bereich übernommen und die Technik des Fingerprinting entwickelt. Im Web Tracking ist das Fingerprinting eine Sammlung mehrerer Methoden, die verschiedene Technologien nutzen, um unterschiedliche Datenpunkte sammeln zu können. Der Fingerprint im Nutzer-Tracking Sinn ist dann einfach eine Zusammenstellung verschiedener Erkennungsmerkmale eines Nutzers bzw. des Browsers des Nutzers. Diese Erkennungsmerkmale werden dann zusammen in einer einzigartigen ID verbunden, die ein Webserver dann nutzen kann, um einen Nutzer auch über mehrere Seiten hinweg zu identifizieren [21]. Die Annahme dabei ist, dass jeder Browser eines Nutzers unterschiedlich zu einem anderen ist. Beim Logfile-Tracking konnte man sehen, dass durch jede Anfrage verschiedene Daten über den Browser, wie z.B. der User Agent mitgesendet werden. Von solchen Daten zieht das Fingerprinting ihren Nutzen und erstellt auf Basis dieser eine ID des Browsers bzw. des Nutzers. Noch geht man davon aus, dass die Genauigkeit des eindeutigen Identifizierens noch nicht die eines menschlichen Fingerabdrucks erreicht hat, aber es konnten schon Studien erstellt werden, die Fingerprinter erstellt haben, welche User mit einer Genauigkeit von 90 % - 99 % identifizieren konnten [118].
Im Vergleich zum Cookie-Tracking oder anderen speicher-basierten Tracking-Methoden bieten sich hier viele Vorteile an. So ist das angesprochene Tracking über mehrere Seiten (auch *cross-site-tracking*) möglich. Es müssen keine Cookies oder andere Speichermöglichkeiten erstellt werden und es wird auch keine Anmeldung bei einer Website benötigt. Dadurch ist der Nutzer komplett unwissend darüber, ob er nun gerade getrackt wird oder nicht [21].
Das Fingerprinting wird in zwei verschiedene Arten unterschieden. Zum einen gibt es das *passive Fingerprinting*, welches schon erwähnt wurde. Hierbei werden die benötigten Browserinformationen ohne Nutzung einer speziellen Anwendung erhoben. Diese Daten kommen dann zum Beispiel aus den Kopfdaten der IP-Pakete oder aber eben auch aus den mitgelieferten Daten bei den HTTP-Anfragen, die ohnehin beim Webserver landen und dann in den Logfiles abgespeichert werden [116]. So können auch einige Daten gesammelt werden und zusammengefügt werden, die dann als eindeutige ID genutzt werden

können.
Auf der anderen Seite steht das *aktive Fingerprinting*, das auch eher die Art des Fingerprinting ist, die flächendeckend für das Nutzer-Tracking genutzt wird. Mit JavaScript-Anwendungen bzw. -Code oder spezielle Plug-ins (Adobe Flash, Microsoft Silverlight, doch beide bald eingestellt) werden Daten vom Browser des Nutzers abgefragt und dann an einen Server weitergesendet [116]. Hierbei können Unmengen an verschiedenen Daten ausgelesen werden, die dann zu einer einzigartigen ID, also dem Digital Fingerprint, zusammengefügt werden können. Neben den gleichen Informationen, die beim passiven Fingerprinting erhoben werden, können Informationen zum Betriebssystem, Bildschirm oder weitere Datenquellen gesammelt werden, was die Genauigkeit der Erstellung einer einzigartigen ID noch besser macht.
Diese beiden Arten des Fingerprinting können dann noch mit bestimmten Fingerprinting-Techniken erweitert werden, die bei den jeweiligen Arten zum Einsatz kommen. Dennoch kann aber nicht jede Fingerprinting-Technik bei beiden Arten angewendet werden, da die beiden Arten, nicht die gleichen Informationen sammeln können, wie schon oben gezeigt. Eine dieser Techniken ist das *Netzwerk und Standort Fingerprinting* [21]. Diese Technik ist eher für das passive Fingerprinting geeignet, denn hier können HTTP-Header ausgelesen werden um Netzwerk- und Standort-Informationen über einen Nutzer erheben zu können. Zu diesen Informationen gehören zum einen die IP-Adresse, der Standort des Nutzers oder andere Netzwerkbasierte Informationen. Man muss aber nicht zwingend den HTTP-Header für diese Informationen auslesen, sondern kann auch mit JavaScript-Anwendungen die IP-Adresse oder den Standort herausfinden. Über die WebRTC-Schnittstelle, eine API zur besseren Kommunikation zwischen Browsern für z.B. verbesserte Telekonferenzen über Web-Anwendungen [119], kann die IP-Adresse herausgefunden werden, was man einen *WebRTC IP Leak* nennt. Mittlerweile versuchen aber Browser diese Schnittstelle für den IP Leak zu schließen, da es zwar Programmiertechnisch ”by design” so gewollt war, dass die IP über die API auslesbar ist, aber die Browserhersteller diese Möglichkeit nicht wirklich willkommen heißen [120]. Auch der Standort ist mit der Geolocation API via JavaScript auslesbar [121].
Das *Device Fingerprinting* ist eine weitere Technik. Hierbei wird versucht einen browserunabhängigen-Identifier durch die Datenerhebung bestimmter Informationen zu genieren, wie es zum Beispiel einer Studie 2019 gelungen ist, bei welcher ein JavaScript-Fingerprinting entwickelt wurden ist, der neben einem Teil der IP-Adresse auch die Betriebssystemversion, Bildschirmauflösung, Zeitzone und weitere Informationen gesammelt hat [21]. Durch diese Auswahl konnte eine ID entwickelt werden, die nicht abhängig von Browser-spezifischen oder Plugin-spezifischen Informationen ist und auch die Mög-

lichkeit des *cross-site-tracking* mit sich bringt [122].
Auch kann man die Technik des *Operating System Fingerprinting* anwenden. Bei dieser Technik werden Betriebssystemspezifische Informationen erhoben und daraus eine ID gebaut. Informationen wie die Systemsprache, Lokale Zeitzone oder Bildschirmgrößen sind hier möglich [21].
Die *Browser Version Fingerprinting*-Technik funktioniert ähnlich, nur dass eben die Browserversion das Ziel der Datenerhebung ist. Über den HTTP-Header die Browserversion zu erheben gilt als eher unzuverlässig und man versucht mit CSS und HTML5 Fingerprinting-Methoden eine zuverlässige Browserversion herauszufinden [21].
Eine sehr beliebte Technik ist das *Canvas Fingerprinting*, was ausschließlich über aktives Fingerprinting funktioniert. Dabei wird das HTML5 *canvas*-Element genutzt, welches ein Bereich auf der Website ist, auf dem per JavaScript Bilder oder Texte gezeichnet werden können. Diese Zeichnungen geschehen im Hintergrund, so dass sie nicht für den Nutzer sichtbar sind. Beim rendern der Zeichnungen gibt der Browser aber detaillierte Informationen zu verschiedenen Dingen, wie der Grafikkarte oder Schriftart [123]. Dazu wird jede Zeichnung mathematisch anders gezeichnet, was nicht unbedingt heißen muss, dass sie visuell anders aussieht. Somit entstehen kleine Unterschiede in den Zeichnungen in verschiedenen Browsern. Daraus kann dann ein Hash-extrahiert werden und dieser als ID für den Browser bzw. den Nutzer angewendet werden [21]. Canvas Fingerprinting ist die mehrheitlich genutzte Fingerprinting-Technik unter den Top 100.000 Alexa Website, wo sie bei mehr als 5,5 % der Webseiten genutzt wird, was zeigt, wie beliebt diese Art ist [21].
Dazu gibt es noch weitere weniger verbreitete Techniken, wie die Nutzung von Browsersuchverläufen, Browserplugins, Browserdimensionen oder Sprachpräferenzen [21].

Die Umsetzung ist je nach Fingerprinting-Technik und Art unterschiedlich. Die Entwicklung eines Fingerprinting-Codes ”from scratch” kann sehr kompliziert und herausfordernd sein. Daher spezialisieren sich auch einige Unternehmen auf Fingeprinting und die Entwicklung dieser Techniken, so z.B. BlueCava (nun Teil von Qualia), ThreatMetrix (auch LexisNexis Risk Solutions) und iovation (Teil von TransUnion). Des Weiteren gibt es online viele Plugins oder Demos zu den verschiedenen Fingerprinting-Techniken [124].
Die einfachste Implementierung von Browser Fingerprinting funktioniert mit dem Plugin *FingerprintJS* [9]. Die Integration erfolgt dabei entweder über ein Content Delivery Network (CDN) oder die Installation per Node Package Manager in einer Node-Anwendung beispielsweise. FingerprintJS ist in der Basis-Version Open-Source und somit kostenlos, doch für Unternehmen gibt es eine kostenpflichtige Version *FingerprintJS Pro*, welche

eine höhere Genauigkeit des Fingerprints bereitstellt und einige weitere Funktionen, die vor allem für Unternehmen interessant sind. Die genauen Unterschiede zwischen der Open-Source- und Pro-Version sind auf der GitHub-Seite des Projekts nachzulesen [9]. Mit Integration des Plugins durch ein CDN, kann ein schnelles *Proof of Concept* wie in Quellcode 10 aufgebaut werden. Hier wird letztendlich die ID des Nutzers nur in der Konsole des Browsers ausgegeben (*console.log*), welcher je nach Browser und Gerät verschieden ist. Um den Nutzer-Tracking-Kreislauf vollenden zu können, muss aber an dieser Stelle des Codes die ID an einen Server weitergegeben werden, damit dieser mit eventuell schon vorhandenen Daten über diesen Nutzer verglichen werden können. Dies kann wieder durch das Page Tagging-Verfahren oder durch andere Techniken, die einen GET- oder POST-Befehl an einen Server triggern, wie zum Beispiel über das noch nicht angesprochene AJAX. AJAX ist in dem Fall eine Möglichkeit HTTP-Anfragen an den Webserver zu stellen, ohne die Website neu laden zu müssen[125].

```
<script>
    function initFingerprintJS() {
        // Initialize an agent at application startup.
        const fpPromise = FingerprintJS.load();

        // Get the visitor identifier when you need it.
        fpPromise
            .then(fp => fp.get())
            .then(result => {
                // This is the visitor identifier:
                const visitorId = result.visitorId;
                console.log(visitorId);
                // pass identifier to server (page tagging, AJAX)
            });
    }
</script>
<script async src="//cdn.jsdelivr.net/npm/@fingerprintjs/fingerprintjs@3/
    dist/fp.min.js" onload="initFingerprintJS()"></script>
```

Quellcode 10: Ausschnitt Implementierung von Fingerprint-Methode durch FingerprintJS, auf Basis von [9]

4.3 Andere Tracking-Arten

Das Kapitel *Andere Tracking-Arten* beschäftigt sich mit allen Tracking-Arten, die nicht klar zum serverseitigen oder clientseitigen Tracking zugeordnet werden konnten. Dabei handelt sich es meistens um Zwischenlösungen beider Arten, die aber in der Praxis

entweder nicht häufig genutzt werden oder (wie beim propriertären ID-Tracking) keine Möglichkeiten bieten, programmiertechnisch die Tracking-Art darzustellen.

Packet Sniffer

Packet Sniffer sind eine Möglichkeit einen Mittelsmann zwischen den beiden Komponenten des Web-Tracking-Prozesses zu installieren. Der Packet Sniffer steht dabei direkt vor dem Webserver und "beschnuppert" regelrecht alle Pakte, die zwischen dem Webserver und dem Client/Browser hin- und hergeschickt werden. Der Packet Sniffer kann dann die Daten herausziehen, die für Analysen benötigt werden.
Diese Möglichkeit kann von Vorteil für Unternehmen sein, die ihre Website auf verschiedene Webserver, ähnlich zu einem privaten CDN, verteilt haben und damit dann auch verschiedene Logfiles entstehen würden. Der Packet Sniffer kann dann dadurch direkt alle Daten, die in die Logfiles fließen würden sofort abgreifen, ohne dass man "händisch" die Daten der jeweiligen Logfiles auf den verschiedenen Webservern zusammenführen muss [1].

Cache-basiertes Tracking

Das *Cache-basierte Tracking* würde eigentlich ein Teil der clientseitigen Tracking-Arten sein, doch durch geringer Nutzung in der Praxis, wird es nun hier bei den übrigen anderen Tracking-Arten mit aufgenommen. Diese Tracking-Methode nutzt verschiedene Zwischenspeicher (Caches), um Nutzer, teils sogar über mehrere Seiten hinweg zu tracken. Eine Cache-Art die dafür genutzt werden kann ist die Gruppe der *Web Caches*. Darunter ist der integrierte Browser Cache eines jeden Browsers zu verstehen, welcher für Tracking-Zwecke genutzt werden kann. Sobald ein Browser eine Datei (HTML, CSS, JavaScript, Bild) von einem Webserver herunterlädt, wird diese im Browser Cache gespeichert, damit beim nächsten Besuch der Website diese schneller geladen werden kann. Dadurch kann eine Website schnell erkennen, ob ein Besucher schon einmal auf der Website war und somit manche Dateien gar nicht mehr geladen werden müssen. Mit der richtigen Logik auf dem Webserver implementiert, kann so ein User identifiziert werden. Vor allem Unternehmen, die Werbung auf verschiedenen Seiten z.B. via Banner-Werbung implementieren können so ganze Browserverläufe nachbilden, wenn auch nur mit den Seiten, auf denen die Banner-Werbung vertreten ist [21].
Der Web Cache kann auf drei Weisen ausgenutzt werden. Zum einen kann man einfach eine ID in einem zwischengespeicherten Dokument einbauen z.B. in einem unsichtbaren *div*-Element in einem HTML-Dokument. Diese ID wird dann benutzt, um den Nut-

zer nachzuverfolgen. Eine weitere Möglichkeit ist die Nutzung von *Loading Performance Tests* via JavaScript zur Nachvollziehung eines erneuten Besuchs einer Website. Die letzte Möglichkeit ist die Ausnutzung von ETags (entity tags) und den *Last-Modified HTTP Headern*. ETags sind IDs, welche jeder Datei, die vom Webserver bereitgestellt wird, beigefügt ist [21]. Durch sie weiß der Webserver, ob die aktuelle Version einer Datei schon geladen wurde oder eine neue Version dessen geliefert werden muss [126]. Diese ID kann nun beschrieben und mit einer selbstgewählten eindeutigen ID belegt werden, der nun mit einer Datenbank an ETag IDs jedes mal verglichen werden kann wenn eine Anfrage mit einem ETag ankommt.
Die beiden weiteren Cache-Gruppen die genutzt werden können sind einmal der DNS Cache und die Gruppe der *Operational Caches*. Während es beim DNS Cache um einen Cache der Lookups von Adressen von Webseiten handelt sind Operational Caches Speicher, die bei Protokollen, wie HTTP oder TLS, zum Einsatz kommen [21].

Reverse Proxies

Reverse Proxies sind einfache Proxy-Server, also Rechner, die zwischen dem Webserver und dem Client installiert werden. Dabei nimmt dieser zuerst alle Anfragen auf, die ein Client sendet und schickt sie dann erst zum Webserver und umgekehrt nimmt er auch erst alle Webserver-Antworten auf. Der Unterschied zum Packet Sniffer liegt daran, dass der Reverse Proxy nicht nur den Datenverkehr lesen, sondern ihn auch verändern kann. Er kann somit jeder Seite, die vom Webserver ausgeliefert wird, noch einen Page Tag hinzufügen und somit die clientseitige Aufgabe vorweg nehmen und überflüssig machen, was ein Vorteil sein kann.
Ein Nachteil ist aber, dass die Reverse Proxies mit hohem Programmieraufwand, wie auch allgemeinen Installationsaufwand, verknüpft sind und deshalb nicht oft für das Tracking und Sammeln von Daten genutzt wird [1].

Evercookies

Evercookies sind keine Cookies im klassischen Sinne. Vielmehr handelt es sich hier um eine JavaScript-API, welche es ermöglicht einen Cookie zu erstellen der ganz allgemein gesprochen für (fast) immer bestehen bleibt. Wenn ein Cookie mit Hilfe von der API gespeichert wird, wird dieser gleichzeitig auf verschiedenen andere Speichermethoden auch gespeichert [127]. Darunter sind die schon vorgestellten Web Storages, Web SQL Database, IndexedDB, aber auch die Flash-Cookies (sogenannte Local Shared Object) und viele weitere Methoden. Eine volle Liste dieser sind im GitHub des Projektes zu

finden [128].
Sobald nun beispielsweise der Cookie gelöscht wurde, wird dies erkannt und es wird sofort aus den Daten in den anderen Speichermethoden der Cookie wieder erstellt. Das gleiche passiert auch, wenn zum Beispiel nur der localStorage gelöscht werden würde. Dann würde einfach nur der localStorage-Eintrag wieder erstellt werden. Man müsste also über ein Dutzend verschiedener Speicher mit dem initial gespeicherten Wert löschen, so dass der "Evercookie" gelöscht ist bzw. die Daten bereinigt sind [127]. Dies macht den Evercookie so hartnäckig. In der Praxis wird diese Form eher weniger angewendet, die Entwickler des Projektes geben auch an, dass der Evercookie nur mit Vorsicht genutzt werden kann, da diese auch Probleme machen können [128]

Cookie Leaks und Cookie Syncing

Nachdem es sich bei den Evercookies nicht notwendigerweise um Cookies gehandelt hat, stehen bei den *Cookie Leaks* bzw. bei den *Cookie Syncing* diese im Mittelpunkt. Einfach gesprochen bedeutet Cookie Leaking, dass Cookies auf einer Domain, einer anderen Domain weitergegeben werden. Dies geschieht dann meistens als Parameter einer Anfrage [21]. Solch eine Technik nutzt z.B. Microsoft, um ihre Cookies, die Daten über Nutzer abspeichern, zwischen ihren verschiedenen Domains (z.B. *bing.com*, *microsoft.com* oder *xbox.com*) hin- und herzureichen. Wenn ein User *microsoft.com* besucht, wird ein Cookie für ihn gesetzt, welcher dann per HTTP-Request an die anderen Domains der Firma Microsoft weitergegeben wird. Sobald nun der User auch *xbox.com* besucht, wird er dort durch den synchronisierten Cookie wiedererkannt. Dieser Mechanismus ist dabei sogar unabhängig davon, ob der Nutzer die zweite Website noch in der gleichen Sitzung besucht oder erst bei einer nächsten. Auch Google kann diese Technik für ihre Online Advertising-Geschäfte nutzen. Durch Cookie Syncing können Dritte Daten austauschen und diese für besser optimierte Kampagnen in der Google Advertising-Umgebung nutzen [129].

ID-Tracking und Cross-Device-Tracking

Ähnlich wie die login-basierten-ID-Systeme, die The Trade Desk bzw. nun Prebid mit der Unified ID 2.0 zum Tracking von Nutzern derzeit entwickeln, gibt es schon funktionierende und implementierte ID-Tracking-Systeme von Konkurrenten. *ID-Tracking* bedeutet in diesem Sinne einfach nur, dass ein Nutzer über eine einzigartige ID über mehrere Webseiten und auch Geräte getrackt werden soll. Diesen Begriff kann man mit dem *Cross-Device-Tracking* verbinden. Diese Tracking-Art beschreibt das Nachverfolgen von

Nutzern über mehrere Geräte und diese funktioniert nur über zwei verschiedene Verfahren. Entweder trackt man jeden User mit einer einzigartigen ID (hier ID-Tracking) oder man trackt über eine Geräte-ID, die ähnlich zum Fingerprinting anhand verschiedener Informationen des Gerätes erstellt wird. Die erste Art wird aber schon flächendeckend genutzt und zwar mit login-basierten-ID-Systemen. Vor allem soziale Netzwerke bedienen sich an diesem Verfahren, in dem sie durch den Login ihrer Nutzer eine einzigartige ID erstellen, die auch auf andere Geräte bei jedem Login mitgenommen wird. Somit können die Nutzer auf jedem Gerät, auf dem sie mit ihrem Social-Media-Account angemeldet sind, nachverfolgt werden [130]. Auch Online-Shops wie Amazon oder die schon stark in das Web Tracking verstrickte Online-Zeitungsbranche nutzen diese Login-Systeme, um ihre User besser tracken zu können. Google geht noch einen Schritt weiter und kann neben der Anmeldung über google.com während Nutzer auf der Suchmaschine suchen auch die Anmeldung im Chrome-Browser per Google Account für noch bessere Nachverfolgung ihrer Nutzer zu ihren Gunsten nutzen [131]. Auch das vorgestellte FLoC-Verfahren würde in diese Sparte fallen. Durch den schon gezeigten hohen Marktanteil des Google Chrome Browsers (61 %) ist das Tracking über dieses Verfahren sehr lukrativ.
Da schon viele Plattformen das ID-Tracking anwenden und Bemühungen von Regierungen und Datenschützern immer mehr die klassischen Tracking-Methoden einschränken, scheint es so, als könnte die Zukunft des Trackings das ID-Tracking basierend auf Logins bei Plattformen sein, was manche auch befürworten, da die Personalisierung des Nutzererlebnis im Internet nach gewissen Meinungen immer an Logins gebunden sein sollen [21].

5 Chancen und Risiken bei der Nutzung von Nutzer-Tracking

Chancen

Um die Chancen (wie auch Risiken) korrekt abwägen zu können, muss man das Nutzer-Tracking von der Website- bzw. Unternehmens- wie auch Nutzerseite betrachten. Die Vorteile der einen Seite können Nachteile der anderen Seite sein und umgekehrt, weswegen diese Isolierung äußerst wichtig ist.
Angefangen mit der Nutzerseite entsteht eine sehr offensichtliche Chance, die auch meistens in dieser Diskussion angewendet wird. Nutzer-Tracking führt zu einem verbesserten Nutzererlebnis. Durch das Tracken können Daten über Nutzer gesammelt werden. Diese können die Unternehmen nutzen, um ihre Nutzer besser zu verstehen. Daraus können dann Maßnahmen definiert werden, wie die Website verändert werden sollte, damit die Nutzer ein besseres Nutzererlebnis (User Experience) erfahren können. Dazu können

auch kleine Änderungen an der Website einen positiven Einfluss haben. So sind Sprachpräferenzen bei vielen Webseiten schon Standard-Prozesse, bei welchen dann je nach Präferenz der Webserver die Website in der korrekten Sprache ausgibt [132].
Des Weiteren können durch das Nutzer-Tracking erhobene Daten für weiterführende Analysen mit Data Mining- bzw. Machine Learning-Modellen genutzt werden, sofern eine datenschutzrechtlich-akzeptierte Zusage durch den Nutzer besteht. Durch sie können nicht nur Webseiten sondern ganze Geschäfte und ihre Angebote verbessert werden, um so im allgemeinen dem potenziellen Kunden, welcher die Website besucht, ein besseres Erlebnis der Marke zu bescheren.
Im speziellen muss hier auch das Thema *Personalisierung* angebracht werden. Diese wird auch durch das Nutzer-Tracking hervorgerufen. Amazon kann beispielsweise die angezeigten Produkte auf ihrer Startseite bei Amazon je nach ihren Vorlieben, die durch das Tracking verschiedener Daten, herausgefunden wurden, anpassen. Dasselbe gilt für die angezeigten Posts in den Social-Media-Feeds bei Instagram, Facebook & Co., die auch personalisiert werden können. Eine Studie des Max-Planck-Instituts hat ergeben, dass Personen in Deutschland per se kein Problem mit Personalisierung haben. Fast 80 % finden, dass die Personalisierung von Restaurant-, Film- und Musikempfehlungen online nicht unbedingt schlecht sind. Doch diese Akzeptanz verliert stark an Rückhalt bei der Personalisierung von Nachrichten aus politischen Kampagnen (39 %) und Beiträgen in Social-Media-Feeds. Vor allem sensible Informationen, wie sexuelle Orientierung und Religion gelten in Deutschland nach der Studie weiter hin als zu heikle Daten, als das man diese für Personalisierungszwecke nutzen könne [133].
Im Vergleich dazu zeigte eine Umfrage aus 2017 durchgeführt von Eurostat, dass 55 % der befragten Deutschen personalisierte Online-Werbung eher negativ entgegentreten [134]. Ein sehr unterschiedliches Bild im Vergleich zur Max-Planck-Studie aus 2020. Nun werden hier zwei verschiedenen Umfragen zweier unterschiedlicher Unternehmen verglichen, die wahrscheinlich verschiedene Umfragemethoden angewendet haben. Dennoch könnte man meinen, dass sich die deutsche Bevölkerung langsam für Personalisierung von Angeboten öffnet. Hier kann eben das Web Tracking helfen, die Personalisierung so korrekt wie möglich zu gestalten, so dass dem Nutzer genau die Filme, Musik oder Restaurants vorgeschlagen werden, die zu ihnen passt.

Auf der Unternehmensseite gibt es vergleichsweise viele Chancen, die durch Web Tracking entstehen. Während auf der Nutzerseite eigentlich nur das Nutzererlebnis sich durch Web Tracking verbessern kann, werden auf der Unternehmensseite viele verschiedene Elemente der Website, wie auch des Geschäfts verbessert. Allen voran steht dort das verbesserte

Nutzerverständnis. Ganz nach dem Zitat „Data is the new oil“ von Clive Humby sind nun einmal Daten heutzutage das, was Öl im letzten Jahrhundert für Standard Oil und & Co. war. Auf den Datensätzen werden heute Geschäftsentscheidungen argumentiert und getroffen. Demnach ist es ein klarer Vorteil für Unternehmen, wenn man mit Hilfe von Web Tracking so viele Daten wie möglich über seine Nutzer sammeln kann. Mit den Daten kann verstanden werden, wer die Nutzer sind. Wo befinden sich die Nutzer, was für ein Verhalten streben sie an, wie reagieren sie auf Berührungspunkte mit dem Unternehmen? Dies können Fragen sein, die durch gewonnene Daten durch Nutzer-Tracking beantwortet werden können. Durch Beantwortung dieser Fragen können dann Entscheidung getroffen werden, sowohl auf der Geschäfts-, wie auch der Websiteebene. Dabei müssen beide Ebenen auch nicht gezwungenermaßen voneinander getrennt sein. Im Bereich der *Conversion Rate Optimization* sind beide Ebenen gefragt. Zum einen werden anhand getrackter Daten Optimierungsmaßnahmen bestimmt, die auf der Website implementiert werden. Zum anderen ist dies aber auch eine Entscheidung auf der Businessebene, denn Conversions sind Verkäufe der Produkte, die sich, wenn man mehr davon verkauft, auch auf das Geschäft im allgemeinen auswirken können. Daher steht hier die Chance des besseren Verständnisses des Nutzers im Vordergrund für Unternehmen.

Dazu können Kosten durch Web Tracking reduziert werden. Insbesondere die Kosten für die Akquirierung eines Kunden können durch besseres Verstehen des Kunden anhand von Daten weiter reduziert werden. Dazu können eventuelle verlorene Kunden wieder gewonnen werden, wenn man weiß, wo diese abgesprungen sind [135]. Durch die Daten lässt sich die sogenannte *Customer Journey* (Kundenreise, Weg des Kunden bis zum Kauf) effizient gestalten und optimieren, so dass das ganze Geschäft davon profitieren kann.

Dazu muss man die Chancen der Nutzer auch als Chance für die Unternehmen wiederverwenden. Erst durch Web Tracking konnte Personalisierung erst so gut werden, wie sie schon heute ist und dabei ist sie immer noch skalierbar. Viele Businessmodelle im Online-Advertising, E-Commerce oder anderer E-Services benötigen das Tracking, um das Nutzererlebnis bestens anzupassen. Viele der Dienste, die wir heute teilweise kostenlos benutzen, könnten ohne das Nutzer-Tracking auf Webseiten eventuell gar nicht existieren. Für Google wäre man eventuell gezwungen sich anzumelden, wenn man nach dem passenden Restaurant sucht und Tracking ohne Logins verboten wäre. Jeglicher Online-Dienst hätte drastisch schlechtere Funktionalität wenn keine Anmeldung vorhanden ist, was zeigt was für eine Chance das Web Tracking bietet, um innovative und nützliche Online-Dienste zu kreieren.

Dazu profitieren vor allem Webseiten, die kostenlose Angebote oder Inhalte anbieten, eben sehr stark vom Web Tracking. Ohne dieses wären kostenlose Inhalte eventuell nicht mehr möglich, wenn keine Umsätze durch Online Advertising, was vor allem durch Web Tracking funktioniert, erwirtschaftet werden [136].

Risiken

Nun stehen aber auf der anderen Seite ein Berg an Risiken den Chancen gegenüber, die aus dem Nutzer-Tracking auf Webseiten resultieren. Zu aller erst muss das Risiko für die Online-Privatsphäre angesprochen werden. Es wurden schon mehrfach in dieser Arbeit gezeigt, warum Tracking eine Gefahr für die Online-Privatsphäre sein kann. Die eigenen Daten in die Hand anderer zu legen ist immer gefährlich. Die schlechte Abspeicherung von Daten, so dass diese durch Hacker oder einfach nur durch Datenlecks gestohlen werden können oder die Weitergabe der Daten an Dritte sind die zwei größten Risiken die Web Tracking mit sich bringen. Durch Datensammlung entsteht eine gewisse Verantwortung, welcher die Website-Betreiber gerecht werden müssen. Daten können auf viele verschiedene Weisen missbraucht werden, wenn sie in den falschen Händen sind. Darum sind immer noch viele besorgt, wie viele Daten, vor allem von großen Unternehmen, gesammelt werden können [137]. Online-Privatsphäre wird mittlerweile von vielen als ein Privileg und Recht angesehen [137], wobei vor allem das Nutzer-Tracking der Online-Privatsphäre ein Dorn im Auge ist. Hier stellt sich insbesondere die Frage, wie Web Tracking gestaltet werden kann, dass es die Online-Privatsphäre respektieren kann und trotzdem effektiv genug für die Unternehmen ist und sich keine Online-Dienste deswegen stark verschlechtern. Bis jetzt konnte aber noch keine wirkliche Lösungen gefunden werden einen Tracking-Mechanismus anzuwenden, der Online-Privatsphäre wirklich respektiert und im Einklang mit dessen arbeitet.
Des Weiteren kann Web Tracking auf Unternehmensseite in gewisser Weise schädlich sein für eine Marke. Immer mehr Prozesse gegen Unternehmen wegen der Nutzung von hinterhältigen Tracking-Methoden kommen auf [138], was verständlicherweise nicht gut für eine Marke ist. Solche Effekte treten erst recht ein, wenn es um die Daten geht, die aus Web Tracking resultieren. Eine Studie von Microsoft in Zusammenarbeit mit iProspect hat ergeben, dass 85 % der Befragten ihre Stellung zu einem Unternehmen geändert haben, nachdem eine Datenleck publik wurde. 65 % haben sogar komplett ihre Beziehungen zu diesen Unternehmen gekappt. Dazu verlieren Nutzer auch das Vertrauen in diese Unternehmen [137]. Ein Datenleck, so wie der neueste von LinkedIn, welcher Information von knapp 92 % der Nutzerschaft freigelegt hat [139], kann dabei stark dem Vertrauen in die Marke LinkedIn schaden.

Dazu ist die Intransparenz des Web Trackings eine Gefahr resultierend aus der Nutzung. Gerade die häufig genutzten Tracking-Methoden, wie das Fingerprinting oder das Cookie-Tracking sind für den User entweder teilweise oder sogar völlig unsichtbar. Das heißt der Nutzer hat überhaupt kein Wissen darüber, ob er gerade getrackt wird und schlussfolgernd daraus, ob er der besuchten Website sein Vertrauen schenken kann. Durch die angesprochenen *Cookie-Notices* und *Cookie-Walls* (welche hauptsächlich wegen der Einführung der DSGVO nun so häufig vorzufinden sind) wurde dies nun etwas besser zu Gunsten der Endnutzer. Trotzdem ist man hier noch weit weg von einer Transparenz der Datensammlung. Durch die Gestaltung dieser Cookie-Banner auf den Websites wird meist schon suggeriert, dass der User das Tracking akzeptieren und sich damit abfinden soll. Psychologische Trigger, wie die Nutzung korrekter Farben auf Button-Elementen im Cookie-Banner helfen, dabei den Nutzer schon zum akzeptieren zu bringen, ohne dass sich dieser mit dem Thema beschäftigen muss. Es wird teilweise enorm stark versucht den User davon abzuhalten sich wirklich mit dem Web Tracking und den Daten, die eine Website von ihm sammelt, zu beschäftigen. Dies ist verständlich von der Unternehmensseite, denn durch die Abfrage, ob Tracking vom Nutzer erlaubt wird, kann schon eine hohe Zahl an Nutzern das Tracking ablehnen, was man im Fall Apple und der Einführung einer Abfrage, ob Tracking via Werbe-ID erlaubt wird, zu sehen war [65].
Zudem spielt auch das Thema *Dritte* bei der Transparenz eine Rolle. Das Third-Party-Tracking wird zunächst noch weiterhin ein Bestandteil der Tracking-Welt bleiben, solange dies eben durch Browserabänderungen nicht mehr möglich sein wird. Dabei ist vor allem das Problem transparent zu zeigen, mit wem, welche Daten und warum geteilt werden. Bei Google beispielsweise ist das ohne umfangreiche Suche überhaupt nicht möglich. Bei einem Online-Magazin wie *spiegel.de* werden alle Partner zwar aufgelistet, wo es in diesem Fall ohnehin schon Unmengen an Drittunternehmen sind (Stichwort Vertrauen), doch fehlt hier eine Erklärung, wer das Drittunternehmen ist und für was konkret die gesammelten Daten genutzt werden. Dies sind zwar keine direkten Anforderungen der DSGVO, dennoch geht Transparenz viel weiter als die Verordnung und sollte unabhängig von dieser versucht werden besser umzusetzen.
Abschließend muss die Datenmacht als Risiko angesprochen werden. Dabei sind vor allem die großen Technologien-Unternehmen, namentlich Google, Amazon, Facebook, Apple und Microsoft (GAFAM) zu behandeln. Wie erwähnt sind Daten nun der wohl wichtigste Rohstoff des 21. Jahrhunderts. ”Big Tech” konnte in den letzten Jahrzehnten große Datensätzen zusammenstellen mit denen sie Algorithmen entwickelten, die präzise vorhersagen, welches Produkt für einen Nutzer genau jetzt das richtige wäre oder was der Nutzer als Werbung auf einer Website sehen muss. Dabei sind eben diese 5 Technologie-

Unternehmen am fortschrittlichsten und haben ganze Märkte, die auf Basis von Daten und demnach auch Webtracking so nur existieren können, übernommen. Diese Datenmacht, die sich diese Technologie-Unternehmen aufgebaut haben können gefährlich für die Marktwirtschaft im allgemeinen sein. Monopole sind immer problematisch, da dadurch Innovation geschwächt wird. Wenn nun einige wenige Technologie-Unternehmen sich weiter Monopole durch ihre Daten aufbauen, werden kleinere Unternehmen untergehen und die Wirtschaft insgesamt geschwächt werden. Dies ist die wirtschaftliche Sicht zu Datenmonopolen, doch kann die Datenmacht auch für den Bürger speziell gefährlich sein. Allen voran eben ist die Gefährdung der Online-Privatsphäre ein Resultat von Datenmacht einiger weniger Technologie-Unternehmen, die insbesondere personenbezogene Daten sammeln und nutzen. Doch auch die Wahl verschiedener Services kann somit eingeengt werden. Beispielsweise könnte ein Service von einem der GAFAM-Unternehmen durch das Datenmonopol bei weitem besser sein, als der gleichartiger Service eines kleineren Unternehmens. Solche Unterschiede herbeigeführt durch Datenmonopole können schon allein im Suchmaschinenmarkt beim Zweikampf der GAFAM-Unternehmen Google und Microsoft (bing) betrachtet werden. Eine Studie hat ergeben, das bei Bing deutlich mehr Falschinformationen in den Suchergebnissen dargestellt werden, als bei Google, was natürlich auch an den Daten liegt, auf denen der Bing-Algorithmus liegt [140]. Dadurch wird die Chance, bessere Services anzubieten völlig ausgeglichen, denn wenn das Sammeln von Daten trotzdem in schlechte Dienste resultiert, kann man hier nicht von einer Chance der Verbesserung von Diensten sprechen. Aus den vorangegangen Gründen sind daher auch Datenmonopole ein weitergehendes Risiko das aus Nutzer-Tracking resultieren kann.

Ein weiteres Risiko resultierend aus Web Tracking ist, dass staatliche Überwachung nochmals viel intensiver ausgeführt werden könnte. Während ohnehin schon auf manchen Staats-Webseiten Tracker eingebaut sind [4], könnten diese das Nutzer-Tracking noch stärker annehmen und ihre Tracking-Initiativen erweitern. Vor allem in totalitären Systemen könnte dies eine weitere Möglichkeit sein, das Volk weiterhin zu unterdrücken. Die Ausnutzung von Web Tracking durch den Staat ist demnach auch nicht auszuschließen und stellt daher auch ein Risiko dar.

6 Fazit

In der vorangegangenen Abhandlung wurde dargestellt was *Nutzer-Tracking auf Webseiten* bedeutet. Es wurde gezeigt, dass Nutzer-Tracking ein Kreislauf ist, der aus den

Schritten Datensammlung, -speicherung und -auswertung besteht. Dazu kann dieser Prozess zudem Namen wie *Web Tracking* oder *Online Tracking* annehmen. Beim Nutzer-Tracking dreht sich alles um die beiden Akteure des Prozesses: Der Nutzer und die Website bzw. der Website-Betreiber. Die Website sammelt dabei Daten vom Nutzer und nutzt sie im Web Tracking-Prozess zu Analysezwecken. Oft im Diskurs dabei steht auch das *Third-Party-Tracking*, bei welchem Tracker von Dritten auf einer Website hinterlegt werden und somit Daten von Nutzern meist in deren Unwissenheit sammelt.
Auch wurde deutlich, wie facettenreich die Nutzung von Nutzer-Tracking ist. Von Online-Werbung, über E-Commerce, bis hin zu der Nutzung durch ganze Staaten und Regierungen reicht die Verwendung der Methodik, wobei diese Auflistung nicht einmal flächendeckend ist. Diese breite Streuung der Nutzung wurde dazu mit Hilfe von Studien belegt und die wirkliche Verbreitung von Tracking, welche mit 71 % eine enorme Reichweite darstellt.
Der Kampf zwischen der Anti-Tracking-Bewegung, in Form von Browserherstellern, Adblocker-Entwickler und auch Staaten, und den Tracking-Anbietern ist an einem Punkt angelangt, an welchem die Thematik immer kritischer für die Tracking-Anbieter aussieht. Die Bemühungen der Anti-Tracking-Bewegungen gängige Tracking-Arten, wie das Cookie-Tracking und das Fingerprinting, weitestgehend einzuschränken, stellen die Tracking-Anbieter vor eine große Herausforderung Third-Party-Tracking und Web Tracking im allgemeinen effizient anwenden zu können. Neue login-basierte-Tracking-Systeme, wie es die *Unified ID 2.0* darstellt, sollen hier die Lösung sein, doch ist das Vertrauen in diese noch nicht allzu hoch. Dasselbe gilt für Bemühungen von Tracking-Giganten, wie Google, die mit ihren neuen Vorschlägen für das Third-Party-Tracking bis jetzt nicht die Gemüter der Anti-Tracking-Bewegung beruhigen konnte. Zusätzlich bringen Staaten die Tracking-Konzerne mit DSGVO, ePrivacy & Co. weiter in die Bredouille, so dass diese auch rechtlich gesehen Schwierigkeiten haben.

Dazu wurden die verschiedenen Arten des Trackings vorgestellt. Während es zum einen die *serverseitigen Tracking-Arten* gibt, welche nur aus dem Logfile-Tracking bestehen und direkt auf dem Webserver Nutzer-Tracking betreiben, gibt es auf der anderen Seite die stark dominierenden *clientseitigen Tracking-Arten*, die wiederum das Tracking auf dem Client, also dem Browser des Nutzers, beschreiben. Hierunter fallen die Tracking-Methoden, wie das *Page Tagging*, das *Cookie-Tracking* oder auch die *Fingerprinting-Methoden*, die beschrieben und mit Codebeispielen näher behandelt wurden. Für das Cookie-Tracking wurden zudem alternative Speichermethoden, wie der *Web Storage* oder die *IndexedDB*, vorgestellt und deren Unterschiede erläutert. Die Vielfalt der Möglichkei-

ten auf der clientseitigen Tracking-Seite sind erstaunlich hoch und zeigen, wie schwierig es ist, einen Überblick über die Tracking-Welt zu behalten. Dies zeigten auch die *anderen Tracking-Arten*, was eine Sammlung aller Tracking-Arten darstellt, die nicht eindeutig zu einem der vorangegangenen Arten gezählt werden konnten.

Zuletzt wurde aufgezeigt, welche Chancen und Risiken bei der Nutzung von Nutzer-Tracking entstehen können und auch schon jetzt entstanden sind. Man konnte erkennen, dass vor allem viele Chancen für Unternehmen für das Geschäft entstehen können. Zwar entstehen auch Chancen und Vorteile für die Nutzer, dessen Daten getrackt wurden, doch wurde deutlich, dass die Chance des besseren Nutzererlebnisses ein zweischneidiges Schwert, für das auf der anderen Seite Gefahren für die Online-Privatsphäre entstehen, was das relevanteste Risiko von Nutzer-Tracking ist. Dazu entstehen vor allem Risiken durch die resultierenden Daten aus der exzessiven Datensammlung mit dem Nutzer-Tracking, die Nutzer stark betreffen. Insbesondere konnte dargelegt werden, dass Risiken hauptsächlich für Nutzer entscheidende und im Vergleich vor allem Chancen für Unternehmen bzw. die Website-Betreiber entstehen. Diese Diskrepanz in der Chancen- und Risiko-Entstehung zeigt, dass betrachtet von der Nutzer-Seite das Nutzer-Tracking eher als risikoschaffend und von der Website-Seite als chancenkreierend bewertet werden kann.

Literatur

[1] M. Hassler, *Digital und Web Analytics - Metriken auswerten, Besucherverhalten verstehen, Website optimieren.* mitp, 4 ed., 2017.

[2] ZAW, "Marktanteile der einzelnen Werbemedien im deutschen Bruttowerbemarkt im Jahr 2019 [Graph], Statista." https://de.statista.com/statistik/daten/studie/154767/umfrage/mediasplit-im-deutschen-werbemarkt/, 2020. [aufgerufen am 16. Mai 2021].

[3] "Government websites: If you are not the product, you're the taxpayer, WhoTracks.me." https://whotracks.me/blog/government_websites_september.html, 2018. [aufgerufen am 17. Mai 2021].

[4] A. Karaj, S. Macbeth, R. Berson und J. M. Pujol, "WhoTracks .Me: Shedding light on the opaque world of online tracking," 2019.

[5] "ELK Stack Tutorial: What is Kibana, Logstash & Elasticsearch?, Guru99." https://www.guru99.com/elk-stack-tutorial.html. [aufgerufen am 26. Juni 2021].

[6] "JavaScript Cookies, W3 Schools." https://www.w3schools.com/js/js_cookies.asp. [aufgerufen am 28. Juni 2021].

[7] "HTML Web Storage API, W3 Schools." https://www.w3schools.com/html/html5_webstorage.asp. [aufgerufen am 1. Juli 2021].

[8] "JavaScript IndexedDB, JavaScript Tutorial." https://www.javascripttutorial.net/web-apis/javascript-indexeddb/. [aufgerufen am 1. Juli 2021].

[9] "Fingerprint.js, GitHub." https://github.com/fingerprintjs/fingerprintjs. [aufgerufen am 2. Juli 2021].

[10] M. Brandt, "EU: 71 % wissen, was Cookies tun, Statista." https://de.statista.com/infografik/7960/umgang-mit-cookies/, 2017. [aufgerufen am 09. Mai 2021].

[11] S. Siebert, "Datenschutz bei Tracking-, Webcontrolling- und Analysetools, E-Recht24." https://www.e-recht24.de/artikel/datenschutz/6203-datenschutz-bei-tracking-webcontrolling-analysetools.html, 2019. [aufgerufen am 09. Mai 2021].

[12] I. Belcic, "A Complete Guide to Web Tracking (and How to Avoid It), Avast." https://www.avast.com/c-web-tracking, 2021. [aufgerufen am 09. Mai 2021].

[13] "Tracking Deutsch Übersetzung, Langenscheidt." https://de.langenscheidt.com/englisch-deutsch/tracking, 2021. [aufgerufen am 13. Mai 2021].

[14] K. Wübbenhorst und W. Krieger, "Tracking - Ausführliche Definition im Online-Lexikon, Gabler Wirtschaftslexikon." https://wirtschaftslexikon.gabler.de/definition/tracking-47221/version-270487, 2018. [aufgerufen am 13. Mai 2021].

[15] "Tracking, Ryte Wiki." https://de.ryte.com/wiki/Tracking. [aufgerufen am 13. Mai 2021].

[16] "Key Performance Indicator (KPI) - Ausführliche Definition im Online-Lexikon, Gabler Wirtschaftslexikon." https://wirtschaftslexikon.gabler.de/definition/key-performance-indicator-kpi-52670/version-275788, 2018. [aufgerufen am 13. Mai 2021].

[17] "User, Ryte Wiki." https://de.ryte.com/wiki/User. [aufgerufen am 13. Mai 2021].

[18] S. Schelter und J. Kunegis, "On the Ubiquity of Web Tracking: Insights from a Billion-Page Web Crawl," 2016.

[19] R. Lackes, M. Siepermann und T. Kollmann, "Website - Ausführliche Definition im Online-Lexikon, Gabler Wirtschaftslexikon." https://wirtschaftslexikon.gabler.de/definition/website-49665/version-272893, 2018. [aufgerufen am 13. Mai 2021].

[20] A. Lerner, A. Kornfeld Simpson, T. Kohno und F. Roesner, "Internet Jones and the Raiders of the Lost Trackers: An Archaeological Study of Web Tracking from 1996 to 2016," Aug. 2016.

[21] T. Bujlow, V. Carela-Español, J. Solé-Pareta und P. Barlet-Ros, "Web Tracking: Mechanisms, Implications, and Defenses," 2015.

[22] N. Al-Fannah, W. Li und C. Mitchell, "Beyond Cookie Monster Amnesia: Real World Persistent Online Tracking," in *ISC*, 2018.

[23] E. Papadogiannakis, P. Papadopoulos, N. Kourtellis und E. P. Markatos, "User Tracking in the Post-cookie Era: How Websites Bypass GDPR Consent to Track Users," 2021.

[24] "Webtracking, Gründerszene Lexikon." https://www.businessinsider.de/gruenderszene/lexikon/begriffe/webtracking/, 2019. [aufgerufen am 13. Mai 2021].

[25] "Mozilla Explains: What is a web tracker?, Mozilla." https://blog.mozilla.org/firefox/what-is-a-web-tracker/, 2019. [aufgerufen am 16. Mai 2021].

[26] T.-C. Li, H. Hang, M. Faloutsos und P. Efstathopoulos, "TrackAdvisor: Taking Back Browsing Privacy from Third-Party Trackers," in *Passive and Active Measurement*, pp. 277–289, 2015.

[27] C. Arthur, "Tech giants may be huge, but nothing matches big data, Guardian." https://www.theguardian.com/technology/2013/aug/23/tech-giants-data, 2013. [aufgerufen am 16. Mai 2021].

[28] K. Hao, "How to poison the data that Big Tech uses to surveil you, MIT Technology Review." https://www.technologyreview.com/2021/03/05/1020376/resist-big-tech-surveillance-data/, 2021. [aufgerufen am 16. Mai 2021].

[29] We Are Social, Hootsuite und DataReportal, "Ranking der größten Social Networks und Messenger nach der Anzahl der Nutzer im Januar 2021 (in Millionen), Statista." https://de.statista.com/statistik/daten/studie/181086/umfrage/die-weltweit-groessten-social-networks-nach-anzahl-der-user/, 2021. [aufgerufen am 16. Mai 2021].

[30] "Online-Werbung Schriftenreihe Wettbewerb und Verbraucherschutz in der digitalen Wirtschaft, Bundeskartellamt." https://www.bundeskartellamt.de/SharedDocs/Publikation/DE/Schriftenreihe_Digitales/Schriftenreihe_Digitales_3.pdf?__blob=publicationFile&v=5, 2018. [aufgerufen am 16. Mai 2021].

[31] Zenith, "Prognose der Entwicklung der Werbeausgaben in den einzelnen Werbemedien weltweit im Zeitraum von 2019 bis 2022 (in Milliarden US-Dollar) [Graph], Statista." https://de.statista.com/statistik/daten/studie/

165828/umfrage/prognose-zu-den-werbeausgaben-weltweit-seit-2010/, 2020. [aufgerufen am 16. Mai 2021].

[32] "Affiliate Marketing einfach erklärt! Definition, Basics & Vergütungsmodelle, OMT." https://www.omt.de/affiliate-marketing/, 2021. [aufgerufen am 16. Mai 2021].

[33] J. Mikians, L. Gyarmati, V. Erramilli, und N. Laoutaris, "Detecting Price and Search Discrimination on the Internet," in *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, HotNets-XI, p. 79–84, Association for Computing Machinery, 2012.

[34] B. Mutschler und F. Eichfeld, *Der erfolgreiche Webauftritt: Kunden gewinnen und binden.* Rheinwerk Verlag, 1 ed., 2016.

[35] D. Jang, R. Jhala, S. Lerner und H. Shacham, "An Empirical Study of Privacy-Violating Information Flows in JavaScript Web Applications," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, p. 270–283, Association for Computing Machinery, 2010.

[36] P. Beuth, "Alles Wichtige zum NSA-Skandal, Zeit Online." https://www.zeit.de/digital/datenschutz/2013-10/hintergrund-nsa-skandal, 2016. [aufgerufen am 17. Mai 2021].

[37] M. Hoppenstedt und W. Wiedmann-Schmidt, "So überwacht der BND das Internet, Spiegel." https://www.spiegel.de/netzwelt/netzpolitik/bundesnachrichtendienst-so-ueberwacht-der-bnd-das-internet-a-216ebe9a-6f22-4883-b1c9-ac5d1442497a, 2020. [aufgerufen am 17. Mai 2021].

[38] E.-M. Weiß, "BKA nutzt WhatsApp-Webfunktion zum Mitlesen bei Verdächtigen, Heise Online." https://www.heise.de/news/BKA-nutzt-WhatsApp-Webfunktion-zum-Mitlesen-bei-Verdaechtigen-4848434.html, 2020. [aufgerufen am 17. Mai 2021].

[39] D. Hipp, "Deutlich mehr BKA-Abfragen zu Internetnutzern, Spiegel." https://www.spiegel.de/netzwelt/web/bundeskriminalamt-deutlich-mehr-bka-abfragen-zu-internetnutzern-a-1261460.html, 2019. [aufgerufen am 17. Mai 2021].

[40] K. Biermann, "BND liefert NSA 1,3 Milliarden Metadaten – jeden Monat, Zeit Online." https://www.zeit.de/politik/deutschland/2015-05/bnd-nsa-milliarden-metadaten, 2015. [aufgerufen am 17. Mai 2021].

[41] S. Riley, "Timeline of Online privacy (1969-2021), VPNExperts." https://www.thevpnexperts.com/vpn/timeline-of-online-privacy/, 2021. [aufgerufen am 17. Mai 2021].

[42] M. Brandt, "Sie wissen, was du letzten Sommer geklickt hast, Statista." https://de.statista.com/infografik/12252/tracking-reichweite-von-internet-unternehmen/, 2017. [aufgerufen am 17. Mai 2021].

[43] F. Manjoo, "I Visited 47 Sites. Hundreds of Trackers Followed Me., The New York Times." https://www.nytimes.com/interactive/2019/08/23/opinion/data-internet-privacy-tracking.html, 2019. [aufgerufen am 18. Mai 2021].

[44] T. Wittenhorst, "Daten hunderter Millionen Facebook-Nutzer erneut im Netz entdeckt, Heise Online." https://www.heise.de/news/Daten-hunderter-Millionen-Facebook-Nutzer-erneut-im-Netz-entdeckt-6005192.html, 2021. [aufgerufen am 18. Mai 2021].

[45] Z. Whittaker, "A huge database of Facebook users' phone numbers found online, Techcrunch." https://techcrunch.com/2019/09/04/facebook-phone-numbers-exposed/, 2019. [aufgerufen am 18. Mai 2021].

[46] P. Bump, "The Death of the Third-Party Cookie: What Marketers Need to Know About Google's Looming Privacy Pivots, HubSpot." https://blog.hubspot.com/marketing/third-party-cookie-phase-out, 2021. [aufgerufen am 20. Mai 2021].

[47] M. Wood, "Today's Firefox Blocks Third-Party Tracking Cookies and Cryptomining by Default, Mozilla Blog." https://blog.mozilla.org/blog/2019/09/03/todays-firefox-blocks-third-party-tracking-cookies-and-cryptomining-by-default/, 2019. [aufgerufen am 21. Mai 2021].

[48] D. Aleksandersen, "How Different Browsers Handle First-Party and Third-Party Cookies, Clearcode." https://clearcode.cc/blog/browsers-first-third-party-cookies/, 2019. [aufgerufen am 20. Mai 2021].

[49] M. Wlosik, "What breaks if you use First-Party Isolation in Firefox, Ctrl Blog." https://www.ctrl.blog/entry/firefox-fpi.html, 2018. [aufgerufen am 20. Mai 2021].

[50] M. Zawadziński, "What Is Intelligent Tracking Prevention and How Does It Work? [versions 1.0 – 2.3], Clearcode." https://clearcode.cc/blog/intelligent-tracking-prevention/, 2020. [aufgerufen am 20. Mai 2021].

[51] "Brave Broswer Feature List." https://brave.com/features/, 2021. [aufgerufen am 20. Mai 2021].

[52] "Gesetz gegen Wettbewerbsbeschränkungen (GWB) § 18 Marktbeherrschung." https://www.gesetze-im-internet.de/gwb/__18.html, 2021. [aufgerufen am 20. Mai 2021].

[53] StatCounter, "Marktanteile der meistgenutzten Browserversionen weltweit im April 2021, Statista." https://de.statista.com/statistik/daten/studie/158095/umfrage/meistgenutzte-browser-im-internet-weltweit/, 2021. [aufgerufen am 20. Mai 2021].

[54] Cookiebot, "Google ending third-party cookies in Chrome." https://www.cookiebot.com/en/google-third-party-cookies/, 2021. [aufgerufen am 20. Mai 2021].

[55] U. Iqbal, S. Englehardt und Z. Shafiq, "Fingerprinting the fingerprinters: Learning to detect browser fingerprinting behaviors," 2020.

[56] C. Cimpanu, "Brave to generate random browser fingerprints to preserve user privacy, ZDNet." https://www.zdnet.com/article/brave-to-generate-random-browser-fingerprints-to-preserve-user-privacy/, 2020. [aufgerufen am 20. Mai 2021].

[57] "Electronic Frontier Foundation Website." https://www.eff.org/, 2021. [aufgerufen am 20. Mai 2021].

[58] "Trackers and scripts Firefox blocks in Enhanced Tracking Protection, Mozilla Support." https://support.mozilla.org/en-US/kb/trackers-and-scripts-firefox-blocks-enhanced-track, 2021. [aufgerufen am 20. Mai 2021].

[59] "Verhindern von websiteübergreifendem Tracking in Safari auf dem Mac, Apple Support." https://support.apple.com/de-at/guide/safari/sfri40732/mac, 2021. [aufgerufen am 20. Mai 2021].

[60] “The Trade Desk arbeitet an einer cookieunabhängigen Version seiner Unified ID, Adzine.” https://www.adzine.de/2020/08/the-trade-desk-arbeitet-an-einer-cookieunabhaengigen-version-seiner-unified-id/, 2020. [aufgerufen am 20. Mai 2021].

[61] “Cookieunabhängige Unified ID 2.0 nimmt Fahrt auf, Adzine.” https://www.adzine.de/2020/10/cookieunabhaengige-unified-id-2-0-nimmt-fahrt-auf/, 2020. [aufgerufen am 20. Mai 2021].

[62] “Prebid und Epsilon bringen ID-Lösung SharedID an den Start, Adzine.” https://www.adzine.de/2020/12/prebid-und-epsilon-bringen-id-loesung-sharedid-an-den-start/, 2020. [aufgerufen am 20. Mai 2021].

[63] “The Trade Desk übergibt die Unified ID 2.0 an Prebid, Adzine.” https://www.adzine.de/2021/02/the-trade-desk-uebergibt-die-unified-id-2-0-an-prebid/, 2021. [aufgerufen am 20. Mai 2021].

[64] T. Weidemann, “Apple will Schutz der Privatsphäre in Apps wie geplant umsetzen, t3n.” https://t3n.de/news/apple-schutz-privatsphaere-apps-1339386/, 2020. [aufgerufen am 20. Mai 2021].

[65] D. Petereit, “Katastrophe mit Ansage: Nur 4 Prozent der iPhone-Nutzer erlauben Werbetracking, t3n.” https://t3n.de/news/katastrophe-werbetracking-ios-1377698/, 2021. [aufgerufen am 20. Mai 2021].

[66] E. Laziuk, “iOS 14.5 Opt-in Rate - Daily Updates Since Launch, Flurry.” https://www.flurry.com/blog/ios-14-5-opt-in-rate-att-restricted-app-tracking-transparency-worldwide-us-daily-latest-update/, 2021. [aufgerufen am 20. Mai 2021].

[67] T. Kleinz, “Googles Cookie-Ausstieg: Streit um "Tracking light", Heise Online.” https://www.heise.de/news/Googles-Cookie-Ausstieg-Streit-um-Tracking-light-5994478.html, 2021. [aufgerufen am 20. Mai 2021].

[68] B. Cyphers, “Google’s FLoC Is a Terrible Idea, EFF.” https://www.eff.org/deeplinks/2021/03/googles-floc-terrible-idea, 2021. [aufgerufen am 20. Mai 2021].

[69] D. Riley, “WordPress will block Google’s new FLoC web tracking cookie replacement, siliconANGLE.” https://siliconangle.com/2021/04/19/wordpress-

will-block-googles-new-floc-web-tracking-cookie-replacement/, 2021. [aufgerufen am 20. Mai 2021].

[70] Carike, "Proposal: Treat FLoC like a security concern, WordPress." https://make.wordpress.org/core/2021/04/18/proposal-treat-floc-as-a-security-concern/, 2021. [aufgerufen am 20. Mai 2021].

[71] "Why Brave Disables FLoC, Brave." https://brave.com/why-brave-disables-floc//, 2021. [aufgerufen am 20. Mai 2021].

[72] A. Sharma, "GitHub disables Google FLoC user tracking on its website, BleepingComputer." https://www.bleepingcomputer.com/news/security/github-disables-google-floc-user-tracking-on-its-website/, 2021. [aufgerufen am 20. Mai 2021].

[73] E.-M. Weiß, "DuckDuckGo blockiert Googles FLoC per Erweiterung, Heise Online." https://www.heise.de/news/DuckDuckGo-blockiert-Googles-FLoC-per-Erweiterung-6011627.html, 2021. [aufgerufen am 20. Mai 2021].

[74] D. Windelband, "DSGVO gilt auch in Norwegen, Island und Liechtenstein, datenschutz notizen." https://www.datenschutz-notizen.de/dsgvo-gilt-auch-in-norwegen-island-und-liechtenstein-4421110/, 2018. [aufgerufen am 22. Mai 2021].

[75] S. Siebert, B. Brünen und L. Lexow, "DSGVO: Das müssen Webseitenbetreiber und Unternehmer über die Datenschutz-Grundverordnung wissen!, eRecht24." https://www.e-recht24.de/datenschutzgrundverordnung.html, 2021. [aufgerufen am 22. Mai 2021].

[76] "Datenschutz-Grundverordnung, Bundesministerium der Justiz und für Verbraucherschutz." https://www.bmjv.de/DE/Themen/FokusThemen/DSGVO/DSVGO_node.htmll, 2021. [aufgerufen am 22. Mai 2021].

[77] "Kommissionsbericht: Die EU-Datenschutzvorschriften stärken die Rechte der Bürgerinnen und Bürger und sind zeitgemäß, Europäische Kommission." https://ec.europa.eu/commission/presscorner/detail/de/ip_20_1163, 2020. [aufgerufen am 22. Mai 2021].

[78] C. Conrad, "Evaluierungsbericht zur DSGVO: Eine erste positive Bilanz?, datenschutz notizen." https://www.datenschutz-notizen.de/evaluierungsbericht-

zur-dsgvo-eine-erste-positive-bilanz-3826322/, 2020. [aufgerufen am 22. Mai 2021].

[79] N. Härting, "Datenschutzgrundverordnung als Instrument der Bevormundung: Trilog erfolgreich, Einwilligung tot, Legal Tribune Online." https://www.lto.de/recht/hintergruende/h/datenschutzgrund-vo-dsgvo-kritik/, 2015. [aufgerufen am 22. Mai 2021].

[80] M. Sweeney und P. Zawiślak, "A Timeline of GDPR Events in AdTech & MarTech, Clearcode." https://clearcode.cc/blog/timeline-gdpr-events-adtech-martech/, 2020. [aufgerufen am 22. Mai 2021].

[81] K. Strauß, "DSGVO & Tracking – Wie ist der Stand im Herbst 2020?, Datenschutzexperte.de." https://www.datenschutzexperte.de/blog/datenschutz-im-unternehmen/dsgvo-tracking-wie-ist-der-stand-im-herbst-2020/, 2020. [aufgerufen am 22. Mai 2021].

[82] "Die ePrivacy-Verordnung und ihr großer Schatten: Womit müssen Sie rechnen?, Ionos." https://www.ionos.de/digitalguide/websites/online-recht/eprivacy-verordnung/, 2021. [aufgerufen am 22. Mai 2021].

[83] H. Schuster, "DSGVO: 48,1 Millionen Euro Bußgelder und kein Ende in Sicht, IT-Business." https://www.it-business.de/dsgvo-481-millionen-euro-bussgelder-und-kein-ende-in-sicht-a-1019909/, 2021. [aufgerufen am 22. Mai 2021].

[84] "Datatilsynets oppgaver, Datatilsynet." https://www.datatilsynet.no/om-datatilsynet/oppgaver/, 2021. [aufgerufen am 22. Mai 2021].

[85] E.-M. Weiß, "Der Foren-Dienst Disqus muss in Norwegen 2,5 Millionen Tracking-Strafe zahlen, Heise Online." https://www.heise.de/news/Der-Foren-Dienst-Disqus-muss-in-Norwegen-2-5-Millionen-Tracking-Strafe-zahlen-6038330.html, 2021. [aufgerufen am 22. Mai 2021].

[86] J. Kulbatzki, "Dating-App Grindr drohen zehn Millionen Euro Strafe, Netzpolitik.org." https://netzpolitik.org/2021/norwegen-dating-app-grindr-drohen-zehn-millionen-euro-strafe/, 2021. [aufgerufen am 22. Mai 2021].

[87] R. Meyers, "Datenschutzgesetze weltweit – eine Übersicht, Versicherungsbetriebe." https://www.versicherungsbetriebe.de/business/2020/

datenschutzgesetze-weltweit---eine-uebersicht-.html, 2020. [aufgerufen am 22. Mai 2021].

[88] We Are Social, Hootsuite, DataReportal, "Anteil der Nutzer von Adblockern unter Internetnutzern in ausgewählten Ländern weltweit im Jahr 2019, Statista." https://de.statista.com/statistik/daten/studie/809634/umfrage/anteil-der-nutzer-von-adblockern-nach-laendern-weltweit/, 2020. [aufgerufen am 22. Mai 2021].

[89] B. Behrens und E. Bałazińska, "Adblocker-Tracking: Wie stark sie Ihre Web Analytics Daten beeinflussen, PIWIK." https://piwikpro.de/blog/wie-stark-adblocker-tracking-und-web-analytics-beeinflussen/, 2018. [aufgerufen am 22. Mai 2021].

[90] S. Rebiger, "Stiftung Warentest testet Tracking-Blocker: Ein Muss für jeden Browser, Netzpolitik.org." https://netzpolitik.org/2017/stiftung-warentest-testet-tracking-blocker-ein-muss-fuer-jeden-browser/, 2017. [aufgerufen am 22. Mai 2021].

[91] M. Hawthorne, "Stop WebRTC From Leaking IP Address with uBlock Origin, Technipages." https://www.technipages.com/stop-webrtc-from-leaking-ip-address-with-ublock-origin, 2020. [aufgerufen am 22. Mai 2021].

[92] D. Berger, "Firefox: uBlock Origin schützt vor versteckten Trackern, Heise Online." https://www.heise.de/newsticker/meldung/Firefox-uBlock-Origin-schuetzt-vor-versteckten-Trackern-4596641.html, 2019. [aufgerufen am 22. Mai 2021].

[93] M. Brinkmann, "Brave browser gets CNAME-based adblocking support, ghacks.net." https://www.ghacks.net/2020/11/17/brave-browser-gets-cname-based-adblocking-support/, 2020. [aufgerufen am 22. Mai 2021].

[94] R. Bisso'o Oyono, "Serverseitiges vs. clientseitiges Websitestracking, Resolution Media." https://resolutionmedia.de/serverseitiges-vs-clientseitiges-websitestracking/, 2020. [aufgerufen am 30. Mai 2021].

[95] "Welche Informationen ein Logfile enthalten kann, Ionos." https://www.ionos.de/digitalguide/online-marketing/web-analyse/logfiles-die-protokolle-der-computerprozesse/, 2016. [aufgerufen am 25. Juni 2021].

[96] "IPv4-Adressen, Elektronik Kompendium." https://www.elektronik-kompendium.de/sites/net/2011211.htm. [aufgerufen am 26. Juni 2021].

[97] "IP-Adressen: Alles, was Sie wissen müssen, Ionos." https://www.ionos.de/digitalguide/server/knowhow/was-ist-eine-ip-adresse/, 2020. [aufgerufen am 26. Juni 2021].

[98] "Importing Apache (httpd) logs into MySQL, StartupCTO." https://www.startupcto.com/server-tech/apache/importing-apache-httpd-logs-into-mysql, 2020. [aufgerufen am 26. Juni 2021].

[99] "CronJob, Ryte Wiki." https://de.ryte.com/wiki/CronJob. [aufgerufen am 26. Juni 2021].

[100] "BigQuery, Google Cloud." https://cloud.google.com/bigquery. [aufgerufen am 26. Juni 2021].

[101] M. Loetzsch, "Server-side tracking: surprisingly easy, Project A." https://insights.project-a.com/server-side-tracking-surprisingly-easy-83d1450cc08f, 2020. [aufgerufen am 26. Juni 2021].

[102] "What's the difference between a cookie, a pixel, and a tag?, Learn Web Analytics." https://learnwebanalytics.com/whats-the-difference-between-a-cookie-a-pixel-and-a-tag/, 2018. [aufgerufen am 26. Juni 2021].

[103] "Page-Tagging: Datenerhebung per Codezeile, Ionos." https://www.ionos.de/digitalguide/online-marketing/web-analyse/page-tagging-datenerhebung-per-codezeile/, 2016. [aufgerufen am 26. Juni 2021].

[104] S. Fiebrandt, "What are cookies? What are the differences between them (session vs. persistent)?, Cisco." https://www.cisco.com/c/en/us/support/docs/security/web-security-appliance/117925-technote-csc-00.html, 2018. [aufgerufen am 28. Juni 2021].

[105] "Document.cookie, MDN Web Docs." https://developer.mozilla.org/en-US/docs/Web/API/Document/cookie. [aufgerufen am 28. Juni 2021].

[106] "Set-Cookie, MDN Web Docs." https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie. [aufgerufen am 28. Juni 2021].

[107] "Using HTTP cookies, MDN Web Docs." https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies. [aufgerufen am 28. Juni 2021].

[108] K. Hartl, "js-cookie, GitHub." https://github.com/js-cookie/js-cookie, 2020. [aufgerufen am 28. Juni 2021].

[109] "Session-Tracking, Ryte Wiki." https://de.ryte.com/wiki/Session-Tracking. [aufgerufen am 2. Juli 2021].

[110] "Web Storage API, MDN Web Docs." https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API. [aufgerufen am 1. Juli 2021].

[111] "Same-origin policy, MDN Web Docs." https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy. [aufgerufen am 1. Juli 2021].

[112] "IndexedDB key characteristics and basic terminology, MDN Web Docs." https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API/Basic_Terminology. [aufgerufen am 1. Juli 2021].

[113] "Verwendung von IndexedDB, MDN Web Docs." https://developer.mozilla.org/de/docs/Web/API/IndexedDB_API/Using_IndexedDB. [aufgerufen am 1. Juli 2021].

[114] S. Belloro und A. Mylonas, "I know what you did last summer: New persistent tracking mechanisms in the wild," vol. 6, pp. 52779–52792, 2018.

[115] "Digital Fingerprinting, ePrivacy." https://www.eprivacy.eu/news/news-detail/article/digital-fingerprinting/, 2020. [aufgerufen am 2. Juli 2021].

[116] "Browser-Fingerprinting: Grundlagen und Schutzmöglichkeiten, Ionos." https://www.ionos.de/digitalguide/online-marketing/web-analyse/browser-fingerprinting-tracking-ohne-cookies/, 2021. [aufgerufen am 2. Juli 2021].

[117] "Fingerabdruck, Duden." https://www.duden.de/rechtschreibung/Fingerabdruck. [aufgerufen am 2. Juli 2021].

[118] D. Goodin, "Now sites can fingerprint you online even when you use multiple browsers, ars Technica." https://arstechnica.com/information-technology/2017/02/now-sites-can-fingerprint-you-online-even-when-you-use-multiple-browsers/, 2017. [aufgerufen am 2. Juli 2021].

[119] "WebRTC API, MDN Web Docs." https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API. [aufgerufen am 2. Juli 2021].

[120] "Can You Get A Users Local LAN IP Address Via JavaScript?, Stackoverflow." https://stackoverflow.com/questions/20194722/can-you-get-a-users-local-lan-ip-address-via-javascript, 2013. [aufgerufen am 2. Juli 2021].

[121] "HTML Geolocation API, W3 Schools." https://www.w3schools.com/html/html5_geolocation.asp. [aufgerufen am 2. Juli 2021].

[122] K. Boda, Á. M. Földes, G. G. Gulyás und S. Imre, "User tracking on the web via cross-browser fingerprinting," vol. 7161, 10 2011.

[123] N. Latto, "What Is Browser Fingerprinting and How Can You Prevent It?, Avast." https://www.avast.com/c-what-is-browser-fingerprinting, 2021. [aufgerufen am 2. Juli 2021].

[124] "GitHub Suche nach "Browser Fingerprinting"-Repositories." https://github.com/search?q=browser+fingerprinting&type=repositories. [aufgerufen am 2. Juli 2021].

[125] "AJAX Introduction, W3 Schools." https://www.w3schools.com/js/js_ajax_intro.asp. [aufgerufen am 2. Juli 2021].

[126] N. Hinternesch, "No Cookies, No Problem — Using ETags For User Tracking, gitconnected." https://levelup.gitconnected.com/no-cookies-no-problem-using-etags-for-user-tracking-3e745544176b, 2020. [aufgerufen am 2. Juli 2021].

[127] J. Ihlenfeld, "Das fast unlöschbare Cookie, Golem." https://www.golem.de/1009/78185.html, 2010. [aufgerufen am 2. Juli 2021].

[128] S. Kamkar, "Evercookie, GitHub." https://github.com/samyk/evercookie. [aufgerufen am 2. Juli 2021].

[129] "Cookie Matching, Google Developers." https://developers.google.com/authorized-buyers/rtb/cookie-guide. [aufgerufen am 3. Juli 2021].

[130] "Cross-Device Tracking, Ryte Wiki." https://de.ryte.com/wiki/Cross-Device_Tracking. [aufgerufen am 2. Juli 2021].

[131] "Tracking Cookies und andere Tracking Methoden, Datenwerk." https://weblog.datenwerk.at/2020/04/02/tracking-cookies-und-andere-tracking-methoden/, 2020. [aufgerufen am 2. Juli 2021].

[132] "Setting language preferences in a browser, W3C." https://www.w3.org/International/questions/qa-lang-priorities, 2011. [aufgerufen am 2. Juli 2021].

[133] A. K. K. Skork, N. Siller, "Wie Deutsche Personalisierung im Internet sehen, Max-Planck-Gesellschaft." https://www.mpg.de/14497641/0221-bild-134137-pm-2020-february, 2020. [aufgerufen am 2. Juli 2021].

[134] M. Brandt, "Personalisierte Werbung? Nein danke!, Statista." https://de.statista.com/infografik/7520/keine-nutzung-personenbezogener-daten-fuer-werbezwecke/, 2017. [aufgerufen am 2. Juli 2021].

[135] J. Pryer, "8 Benefits of Website Analytics & Tracking, Spotler." https://spotler.co.uk/blog/6-key-benefits-of-website-analytics/, 2019. [aufgerufen am 2. Juli 2021].

[136] W. Melicher, M. Sharif, J. Tan, Lu. Bauer, M. Christodorescu und P. Leon, "(Do Not) Track Me Sometimes: Users' Contextual Preferences for Web Tracking," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, pp. 135 – 154, 2016.

[137] "In Brands We Trust - 2020 Consumer Privacy and Brand Trust Research, Microsoft Advertising und iProspect," 2020.

[138] J. Stempel, "Google faces $5 billion lawsuit in U.S. for tracking 'private' internet use, Reuters." https://www.reuters.com/article/us-alphabet-google-privacy-lawsuit-idUSKBN23933H, 2020. [aufgerufen am 3. Juli 2021].

[139] B. Lovejoy, "LinkedIn breach reportedly exposes data of 92 % of users, including inferred salaries [U], 9To5Mac." https://9to5mac.com/2021/06/29/linkedin-breach/, 2021. [aufgerufen am 3. Juli 2021].

[140] D. Bush, A. Zaheer, "Bing's Top Search Results Contain an Alarming Amount of Disinformation, Stanford University." https://fsi.stanford.edu/news/bing-search-disinformation, 2019. [aufgerufen am 3. Juli 2021].