

Konzeption und Entwicklung eines Selenium-basierten cookie-crawlers

Zur Erfassung, Verarbeitung und Analyse von cookie-bannern sowie
dazugehörigen cookies in auf Shopify basierenden online-shops

MASTERARBEIT

ausgearbeitet von

Marvin Nicholas Hallweger

zur Erlangung des akademischen Grades
MASTER OF SCIENCE (M.Sc.)

vorgelegt an der
TECHNISCHEN HOCHSCHULE KÖLN
CAMPUS GUMMERSBACH
FAKULTÄT FÜR INFORMATIK UND
INGENIEURWISSENSCHAFTEN

im Studiengang
INFORMATIK COMPUTER SCIENCE

Erster Prüfer: Prof. Dr. Lutz Köhler
Technische Hochschule Köln

Zweite Prüferin: Prof. Dr. Dietlind Zühlke
Technische Hochschule Köln

Gummersbach, im Mai 2022

Adressen: Marvin Nicholas Hallweger
Hermelinweg 9
58553 Halver
marvin_nicholas.hallweger@smail.th-koeln.de

Prof. Dr. Lutz Köhler
Technische Hochschule Köln
Institut für Informatik (INF)
Steinmüllerallee 1
51643 Gummersbach
lutz.koehler@th-koeln.de

Prof. Dr. Dietlind Zühlke
Technische Hochschule Köln
Institut für Data Science, Engineering, and Analytics (IDE+A)
Steinmüllerallee 6
51643 Gummersbach
dietlind.zuehlke@th-koeln.de

Kurzfassung

Die am 25. Mai 2018 in Kraft getretene Datenschutzgrundverordnung (DSGVO) vereinigt seitdem verschiedene Regelungen für die Verarbeitung personenbezogener Daten sowohl im privaten als auch im öffentlichen Raum. Diese Regelungen bestehen hierbei vereinheitlicht im gesamten EU-Raum.

Das Ziel der Forschung der vorliegenden Arbeit ist die Entwicklung eines Selenium basierten Crawlers zur Erfassung und Analyse von Cookies, Cookie-Bannern sowie der allgemeinen DSGVO-Konformität von deutschen Shopify Online-Shops. Dazu werden die Folgenden Forschungsfragen gestellt.

- Wie und mit welchen Mitteln informieren heutige Shopify-online-shops ihre Kunden über die Nutzung von Cookies? Ist dieses Vorgehen stets DSGVO-konform?
- Ist es möglich, mit Hilfe des open-source-testing-frameworks „Selenium“ einen präzisen Crawler zu implementieren, welcher genaue Angaben über die verschiedenen Cookie-Zustände sowie über die erkennbaren Cookie-Banner in einem Shopify-Online-Shop macht?
- Wie präzise und zuverlässig kann ein Crawler bezüglich dieser verschiedenen Cookie-Aktivitäten Resultate generieren, sodass auf Basis dieser Erkenntnisse ein größeres Datenset an Shopify-Shop-Domains analysiert werden kann?

Um die Forschungsfragen zu beantworten, wurde der entsprechende Crawler entwickelt und die damit generierten Resultate durch verschiedene manuelle Tests validiert. Im Anschluss wurde mit Hilfe des Crawlers ein größeres Datenset von deutschen Shopify Online-Shop-Domains auf deren DSGVO-Konformität in Form einer Hauptanalyse analysiert.

Die Auswertung dieser Hauptanalyse zeigt eine erhebliche Missachtung der verschiedenen Regulierungen der aktuellen Datenschutz-Rechtslage bei einem bestimmten Prozentsatz der analysierten Online-Shops. Diese Missachtungen betreffen die ungenehmigte Nutzung von Third-Party-Cookies (häufig bereits beim Betreten des Online-Shops), die nicht-Respektierung der verschiedenen Nutzerentscheidungen bezüglich des angezeigten Cookie-Banners, die ungenehmigte Verwendung von personenbezogenen Daten sowie die Missachtung der allgemeinen DSGVO-Konformität.

Abstract

The General Data Protection Regulation (GDPR), which took effect on May 25, 2018, has since unified various regulations for the processing of personal data in both the private and public sectors. These regulations exist here unified in the entire EU area.

The goal of the research of this thesis is the development of a Selenium-based crawler for the collection and analysis of cookies, cookie banners and the general GDPR-compliance of German Shopify online stores. For this purpose, the following research questions are posed.

- How and with what methods do today's Shopify online stores inform their customers about the use of cookies? Is this approach always GDPR-compliant?
- Is it possible to implement a precise crawler using the open-source-testing-framework "Selenium", which gives exact information about the different cookie states as well as about the recognizable cookie banners in a Shopify online store?
- How accurately and reliably can a crawler generate results regarding these various cookie activities so that a larger data set of Shopify store domains can be analyzed based on these insights?

To answer the research questions, the corresponding crawler was developed and the results generated with it were validated through various manual tests. Afterwards, the crawler was used to analyze a larger data set of German Shopify online store domains for their GDPR-compliance using a main analysis.

The evaluation of this main analysis shows a significant disregard for the various regulations of the current data protection legal situation in a certain percentage of the analyzed online stores. These disregards concern the unauthorized use of third-party cookies (often already when entering the online store), the non-respect of the various user decisions regarding the displayed cookie banner, the unauthorized use of personal data as well as the disregard of the general GDPR-compliance.

Inhaltsverzeichnis

Abbildungsverzeichnis	i
Tabellenverzeichnis	ii
Abkürzungsverzeichnis	iii
1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung	1
1.3 Forschungskonzept	2
1.4 Eingrenzungen	2
1.5 Gliederung der Arbeit	2
2 Grundlagen und Hintergrund	3
2.1 Datenschutz-Grundverordnung (DSGVO)	3
2.2 Cookie-Technologien	6
2.2.1 Essenzielle Cookies	6
2.2.2 Third-Party-Cookies	7
2.2.3 Browser-Verwaltung der vorhandenen Cookies	7
2.3 Shop-Systeme	8
2.4 Shopify	11
2.4.1 Shopify-App-Store	11
2.4.2 Externe Anbindungen	12
2.4.3 Identifizierung und Erfassung von Shopify-Shops	12
2.5 Reguläre Ausdrücke	14
2.6 Selenium	14
2.7 Webdriver	15
2.8 Python als Grundlage	16
3 Zusammenhängende Arbeiten	17
4 Entwurf und Implementierung	19
4.1 Funktionsweise der Anwendung	19
4.2 Banner-Identifizierung	21
4.2.1 Cookie-Banner Keywörter	22
4.2.2 Screenshot-Funktionalität	24
4.2.3 Fehlgeschlagene Methoden	24
4.3 Cookie-Identifizierung	25
4.4 Online-Shop Datei-System	26
4.5 Identifizierung von Metadaten	27

Inhaltsverzeichnis

4.6	Simulation der Nutzer-Interaktion	28
4.7	Generierung von benötigten Online-Shop-Domains	29
4.8	Analyse der erfassten Online-Shop-Domains	31
4.9	Integrierung der Online-Shop-Domains in die Anwendung	31
5	Erzeugung und Evaluation	32
5.1	Validierung der entwickelten Anwendung	32
5.1.1	Manuelle Überprüfung eines vordefinierten Datensets	32
5.1.2	Wie präzise arbeitet die entwickelte Anwendung?	34
5.2	Analyse der Berechnungsgeschwindigkeiten	35
5.3	Anpassung der Ausführungsmethoden und der Anwendungsstruktur	36
5.4	Erzeugung und Nutzung eines größeren Datensets	37
5.5	Extraktion relevanter Informationen	39
5.6	Resultierende Ergebnisse	41
5.7	Probleme	48
6	Diskussion	50
6.1	Anwendung	50
6.2	Präzision und Genauigkeit	52
6.3	Ergebnisse	53
7	Fazit	54
7.1	Ausblick	55

Literaturverzeichnis

Anhang

Abbildungsverzeichnis

2.1	DSGVO- sowie TTDSG-konformer Cookie-Banner	5
2.2	Google Chrome Cookie-Verwaltung	8
2.3	SaaS vs. PaaS vs. On Premises	10
2.4	Die Browser-Erweiterung „Koala Inspector“	13
2.5	Shopify IP-Owner/IP-Bereich-Suche auf myip.ms	14
2.6	Selenium Webdriver - Beispielhafte Interaktion mit Web-Elementen	15
4.1	Beispielhafte Generierung einer Speicherdatei über das main.py-Skript .	19
4.2	Extrahierung und Analyse über das ExtractData.py-Skript	21
4.3	Identifizierung des „Akzeptieren“-Knopfes in einem Cookie-Banner . . .	28
4.4	Erfassung des korrekten XPaths eines gesuchten Web-Elements	29
4.5	Shopify IP-Owner/IP-Bereich-Detailsuche auf myip.ms	30
5.1	Ermittelte Erkennungspräzision vor/nach Integrierung der Ergebnisse .	34
5.2	Integrierung des Headless-Modes	36
5.3	Nicht erreichbare und analysierbare Online-Shops	40
5.4	Online-Shops, welche DSGVO-konforme Cookie-Banner verwenden . .	42
5.5	Shop-Anzahl: Kann der Nutzer den Cookie-Banner ablehnen?	42
5.6	Shop-Anzahl: Cookie-Quellcode aber kein visualisierter Cookie-Banner .	43
5.7	Shop mit vorhandenen Cookie-Merkmalen und fehlendem Cookie-Banner	43
5.8	Shop-Anzahl: Laden von Third-Party-Cookies zu Beginn	44
5.9	Ungenehmigte geladene Third-Party-Cookies	45
5.10	Anzahl der Online-Shops, welche die Nutzerentscheidungen respektieren	45
5.11	Shop-Anzahl: Cookie-Mengen vor und nach der Akzeptierung	46
5.12	Shop-Anzahl: Hinzufügen der jeweiligen Cookie-Mengen nach Akzeptierung	47
5.13	Anzahl der Online-Shops, welche DSGVO-konform sind	47
5.14	Identifizierung von nicht aufrufbaren Online-Shops	48

Tabellenverzeichnis

2.1	Personenbezogene Daten	4
2.2	Auflistung: Third-Party-Cookies	7
4.1	GDPR Keywörter	22
4.2	Positive Keywörter	22
4.3	Negative Keywörter	23
5.1	Berechnungsgeschwindigkeiten - Test 1	35
5.2	Berechnungsgeschwindigkeiten - Test 2	37

Abkürzungsverzeichnis

DSGVO = Datenschutz-Grundverordnung
GDPR = General Data Protection Regulation
SaaS = Software as a Service
PaaS = Platform as a Service
B2B = Business-to-Business
B2C = Business-to-Consumer
D2C = Direct-to-Consumer
Regex = Regular Expression (Regulärer Ausdruck)

1 Einleitung

1.1 Motivation

Basierend auf der voranschreitenden Digitalisierung gelangt das Thema des Datenschutzes immer weiter in den Vordergrund. Menschen fordern einen besseren Schutz sowie einen gewissenhafteren Umgang mit ihren Daten. Seit der Einführung der Datenschutz-Grundverordnung (DSGVO) im Jahr 2018 sowie den damit verbindlichen Regulierungen für Unternehmen, wird dieses Problem vereinheitlicht in der gesamten EU behandelt. Grundlegend stellt sich nun die Frage, ob sich heutige Unternehmen an diese Regulierungen halten, die Entscheidungen (besonders bei digitalen Produkten) von Nutzern respektieren und personenbezogene Daten von Menschen gewissenhaft verarbeiten.

In dieser Arbeit wird hierfür ein Crawler entworfen und entwickelt, welcher auf dem Testing-Framework Selenium basiert und ein gegebenes Datenset an deutschen Shopify Online-Shop-Domains auf deren DSGVO-Konformität analysiert und bewertet. Der Crawler analysiert hierbei den geladenen Cookie-Banner sowie die verschiedenen Cookies, welche im Laufe des Analyseprozesses des Online-Shops in den Nutzerspeicher integriert werden. Ein zusätzliches Skript wird den Crawler mit einer Kategorisierung der generierten Informationen, sowie mit einer passenden Ausgabe unterstützen.

1.2 Zielsetzung

Das Ziel dieser Arbeit ist die Implementierung einer Selenium-basierten Analyse-Anwendung zur Identifizierung von nicht-DSGVO-konformen deutschen Shopify Online-Shops. Basierend auf diesen Informationen und einer allgemein größeren Analyse (Hauptanalyse) von einer größeren Menge an Online-Shop-Domains, soll hier ein Abbild der allgemeinen DSGVO-Konformität von deutschen Shopify Online-Shops erstellt werden. Die Hauptanalyse basiert hierbei auf den Erkenntnissen von vorlaufenden kleineren Analysen, um eine möglichst große Erkennungspräzision der Anwendung garantieren zu können.

Bei der durchgeführten Hauptanalyse durchläuft der Crawler die einzelnen Online-Shop-Domains, analysiert die einzelnen Online-Shops auf deren DSGVO-Konformität und speichert die ermittelten Informationen in einzelnen Speicherdateien ab. Die Informationen der Speicherdateien werden schlussendlich über ein zusätzlich implementiertes Skript extrahiert, kategorisiert und ausgegeben.

1.3 Forschungskonzept

Das allgemeine Forschungskonzept dieser Arbeit basiert auf dem Entwurf und der Entwicklung des Crawlers. Die aufgestellten Forschungsfragen beziehen sich auf die vorausschauende Analyse von kleineren Mengen an Online-Shop-Domains sowie einer (und damit zusammenhängend) möglichst großen Erkennungspräzision des Crawlers.

- Wie und mit welchen Mitteln informieren heutige Shopify Online-Shops ihre Kunden über die Nutzung von Cookies? Ist dieses Vorgehen stets DSGVO-konform?
- Ist es möglich, mit Hilfe des open-source-testing-frameworks „Selenium“ einen präzisen Crawler zu implementieren, welcher genaue Angaben über die verschiedenen Cookie-Zustände sowie über die erkennbaren Cookie-Banner in einem Shopify Online-Shop macht?
- Wie präzise und zuverlässig kann ein Crawler bezüglich dieser verschiedenen Cookie-Aktivitäten Resultate generieren, sodass auf Basis dieser Erkenntnisse ein größeres Datenset an Shopify Shop-Domains analysiert werden kann?

1.4 Eingrenzungen

Für diese Arbeit wurden aus Zeitgründen einige wenige Eingrenzungen vorgenommen, um den allgemeinen Fokus auf den Entwurf und die Entwicklung des Crawlers sowie die Generierung und Interpretation der Resultate zu legen.

Die erste Eingrenzung beschreibt hierbei den Fakt, dass die entwickelte Anwendung lediglich deutsche Online-Shops analysieren kann. Das entwickelte Keywort-System zur Erfassung der ausschlaggebenden Keywörter hätte bei einem mehrsprachigen Einsatz zu einer deutlichen Expandierung des benötigten Codes geführt.

Zusätzlich wurden lediglich die zehn größten und bekanntesten Online-Shop-Third-Party-Cookies für die durchzuführende Hauptanalyse verwendet. Diese reichten für die Ziele dieser Arbeit vollkommen aus. Bei Bedarf kann diese Menge innerhalb der Anwendung beliebig erweitert werden.

1.5 Gliederung der Arbeit

Neben dieser Einleitung umfasst der restliche Teil dieser Arbeit die Folgende Gliederung. Kapitel 2 Grundlagen und Hintergrund, Kapitel 3 Zusammenhängende Arbeiten, Kapitel 4 Entwurf und Implementierung, Kapitel 5 Erzeugung und Evaluation, Kapitel 6 Diskussion sowie Kapitel 7 Fazit.

2 Grundlagen und Hintergrund

Die Folgenden Kapitel sollen ein Grundwissen bezüglich des Datenschutzes sowie verschiedener Cookie-Technologien verschaffen. Zudem werden unterschiedliche Werkzeuge für die Erstellung von Online-Shops sowie für die Implementierung der zu entwickelnden Anwendung genannt und näher erläutert.

2.1 Datenschutz-Grundverordnung (DSGVO)

Die Datenschutz-Grundverordnung (DSGVO, englisch: General Data Protection Regulation (GDPR)) ist im Mai 2016 in Kraft getreten und gilt offiziell seit dem 25. Mai 2018 in Deutschland und ganz Europa. Die zweijährige Umstellung diente hierbei als Orientierung und Anpassung der Unternehmen. Verbindlich wurde die DSGVO erst nach Ablauf dieser zweijährigen Frist.

Die DSGVO reguliert das allgemeine Datenschutzrecht sowie den unternehmerischen Umgang mit personenbezogenen Daten (im Folgenden Unterkapitel erläutert). Diese Regulierung gilt als einheitliches Gesetz für jeden Mitgliedsstaat der europäischen Union (EU). Anders als die vorherigen Regulierungen der einzelnen Mitgliedsstaaten, gilt die DSGVO nicht nur für Unternehmen der EU, sondern auch für solche, welche Daten von europäischen Bürgern verarbeiten oder eine zusätzliche Niederlassung in der EU besitzen. Das damalige Ziel, größere amerikanische Internet-Konzerne wie facebook und google zu regulieren, wurde durch diese neue DSGVO erfüllt. Kleinere Unternehmen jedoch erfuhren eine größere und zusätzliche Geschäftseingreifende Belastung in Form von Regelungen und Einschränkungen.

Bei einer nicht-Erfüllung der verschiedenen DSGVO-Auflagen, können durch Aufsichtsbehörden Bußgelder von bis zu 20 Mio. Euro aufgestellt werden. Bei weltweit agierenden Unternehmen können Bußen von bis zu 4% des weltweiten Vorjahresumsatzes anfallen. Zudem können (vor allem bei kleineren Unternehmen) durch Kanzleien (häufig auch spezialisiert auf diesen Bereich) Abmahnungen erfolgen, welche wiederum zu teuren Gerichtsverfahren führen können.(Wolford, 2022) (Haucke, 2022a)

Personenbezogene Daten

Personenbezogene Daten sind Daten, welche nicht vollständig anonym erhoben werden, sondern einer bestimmten Person zugeordnet werden können. Die Folgende Tabelle listet einige dieser personenbezogenen Daten auf.

Tabelle 2.1: Beispiele für Personenbezogene Daten

Vorname, Name	Anschrift
E-Mail-Adressen	Geburtsdatum
Geburtsort	Gesundheitsdaten
Fotos	Ausbildung/Beruf
Einkommen	Kauf- Surf- und Klickverhalten

Anzumerken hierbei ist die Tatsache, dass das Kauf- Surf- und Klickverhalten auf den jeweiligen IP-Adressen des Nutzers basiert, die Verfolgung also durch diese garantiert werden kann. (Usercentrics, 2019) (Haucke, 2022a)

Kriterien für einen DSGVO-konformen Cookie-Banner

- Sobald der Nutzer eine Online-Präsenz betreten hat, soll dieser umgehend in der Lage sein, eine entsprechende Entscheidung bezüglich der zu ladenden benötigten Cookies zu treffen. Hierbei dürfen keine nicht essenziellen Cookies im Vorhinein geladen werden. Der Nutzer muss zudem durch einen statischen Cookie-Banner dazu „gezwungen sein“, eine Entscheidung zu treffen, bevor dieser mit der Nutzung der Online-Präsenz fortfährt.
- Wie bereits erwähnt, dürfen nur Cookies geladen werden, welche essenziell für den Betrieb der Online-Präsenz sind. Andere Arten von Cookies (Third-Party-Cookies, auch Drittanbieter-Cookies) benötigen die klare Zustimmung des Nutzers, bevor diese geladen werden dürfen (wird im Laufe der nächsten Kapitel näher erläutert).
- Der Nutzer allein besitzt die einzige Entscheidungsfreiheit, die entsprechenden Cookies abzulehnen oder zu akzeptieren. Hierbei genügen einfache „Annehmen“ oder „Ablehnen“-Knöpfe für eine DSGVO-Konformität nicht aus. Der Nutzer soll eine anschauliche Darstellung der verwendeten Cookie-Arten einsehen können, hierdurch soll eine präzisere Entscheidung garantiert werden. Die verschiedenen Auswahlmechanismen dürfen beim Betreten der Online-Präsenz nicht im Vorhinein ausgewählt sein, der Nutzer muss diese eigenständig auswählen und akzeptieren.

Dieser Auswahlmechanismus soll eine Ergänzung für die bereits vorhandenen „Annehmen“ und „Ablehnen“-Knöpfe sein. Für eine vollständige DSGVO-Konformität sind diese Knöpfe ebenfalls verpflichtend. (Haucke, 2022a) (Intersoft Consulting, 2018)

Erweiterung der DSGVO - Das TTDSG

Das deutsche Telekommunikation-Telemedien-Datenschutz-Gesetz (TTDSG) gilt seit dem 1. Dezember 2021 und ist eine entsprechende Erweiterung (Verbesserungen und Überarbeitungen) der DSGVO. Anders als die DSGVO selbst, war dieses Gesetz nicht nach wenigen Jahren, sondern sofort gültig. Die Folgenden Aspekte beschreiben die wichtigsten Erweiterungen des TTDSG.

- Der negative Knopf (meistens „Ablehnen“ o.Ä) muss identisch mit dem positiven (meistens „Annehmen“ o.Ä) Knopf sein, er darf also nicht mehr in entsprechenden Bannern „versteckt“ sein.
- Es dürfen keine Daten mehr von minderjährigen kommerziell verwendet werden.
- Ergänzend zu den vorher genannten DSGVO-Auswahlmechanismen, muss dem Nutzer zusätzlich erläutert werden, wofür die verschiedenen geladenen Cookies verwendet werden, dies kann in Textform geschehen.
- Das TTDSG beschreibt eine unmittelbare Ausweitung auf Mobile-Apps und Messenger-Dienste, bei welchen die vorher genannte DSGVO noch nicht vollständig gültig war.

Abbildung 2.1 zeigt einen „optimalen“ DSGVO sowie TTDSG-konformen Cookie-Banner. Eine Online-Präsenz, welche diesen (oder einen ähnlichen) Cookie-Banner verwendet und zudem die Entscheidungen des Nutzers bezüglich der Auswahl der zu ladenden Cookies beachtet und respektiert, ist somit DSGVO-konform.

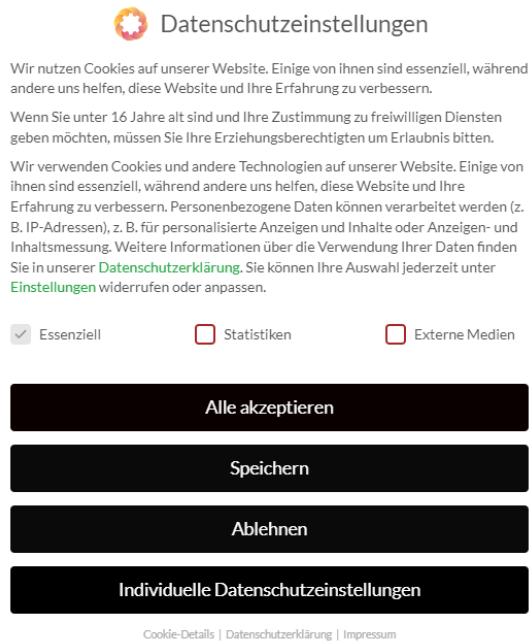


Abbildung 2.1: DSGVO- sowie TTDSG-konformer Cookie-Banner - carlosiebert.de

Das neue TTDSG ermöglicht zudem neue zusätzliche „Schlupflöcher“, mit welchen Kanzleien noch einfacher Abmahnungen an Unternehmen stellen können. In entsprechenden Szenarien werden Unternehmen hierbei nicht nur für Verstöße bezüglich der DSGVO, sondern auch für Verstöße der TTDSG abgemahnt. (Rauner, 2021) (Haucke, 2022b) (Lehmann, 2022)

2.2 Cookie-Technologien

Cookies sind Textdateien, welche von einer Webseite in den Internetbrowser-Speicher eines Nutzers abgelegt und gespeichert werden können. Diese Textdateien dienen zur Speicherung von Nutzer- und Browserinformationen. Im Folgenden werden diese verschiedenen Kategorien von Cookies dargestellt und erläutert. (Haucke, 2022a)

2.2.1 Essenzielle Cookies

Essenzielle Cookies sind Cookies, welche für das Betreiben einer Online-Präsenz wie z.B eines Online-Shops benötigt werden. Hierbei können beispielsweise die sogenannten „Session Cookies“ genannt werden, welche kurzzeitig Informationen wie z.B den derzeitigen Warenkorbinhalt in einem Online-Shop beinhalten. Diese Informationen bzw. diese Cookies werden aus dem Browser-Speicher entfernt, sobald der Nutzer den jeweiligen Browser schließt. Als Erweiterung für diese kurzfristigen „Session Cookies“ können die dauerhaften „persistenten Cookies“ genannt werden. Diese Art von Cookies ist ebenfalls essenziell für den optimalen Betrieb einer Online-Präsenz. Die Besonderheit bei diesen Cookies ist die dauerhafte Speicherung von Informationen, auch nach dem Beenden des Browsers. Cookies dieser Art speichern bspw. die Entscheidungen des Nutzers bezüglich eines angezeigten Cookie Banners. Hierdurch soll verhindert werden, dass sich der Nutzer erneut mit einem Cookie-Banner befassen muss.

Da diese Art von Cookies essenziell für den Betrieb einer Online-Präsenz ist, hat ein Nutzer keine Entscheidungsfreiheit über das Laden dieser Cookies, diese werden also ohne Zustimmung geladen. (BW, 2021)

2.2.2 Third-Party-Cookies

Das Gegenstück zu den essenziellen Cookies sind die sogenannten Third-Party-Cookies (auch Drittanbieter-Cookies). Diese Cookies sind technisch nicht notwendig und dienen hauptsächlich zur Identifizierung sowie zur Nachverfolgung bezüglich des Surfverhaltens von Nutzern im Internet. Diese Nachverfolgung basiert hierbei auf der Erfassung von personenbezogenen Daten von Nutzern.

Anders als bei den essenziellen Cookies, wird bei dieser Art von Cookies eine eindeutige Einwilligung des Nutzers benötigt, bevor diese in den Browser-Speicher integriert werden dürfen. (BW, 2021)

Häufige Third-Party-Cookies in Online-Shops

Im Folgenden werden einige Third-Party-Cookies aufgelistet, welche häufig in Online-Shops eingesetzt und aktiv zur Identifizierung von Nutzern verwendet werden.

Tabelle 2.2: Auflistung von häufig genutzten Third-Party-Cookies

Cookie-Name	Unternehmen/Übersetzung	Verwendungszweck
_fbp	Facebook Pixel (Facebook)	Nutzeridentifizierung
_pin_unauth	Pinterest Tag (Pinterest)	Nutzeridentifizierung
_ga, _gat, _gid	Google Analytics (Google)	Nutzeridentifizierung
_gcl_au	Google Adsense (Google)	Nutzeridentifizierung
_hjIncludedInPageviewSample	Hotjar	User Experience Tool
__hssc, hubspotutk	HubSpot	CRM-System
__kla_id	Klaviyo	E-Mail-Marketing
_lfa	Leadfeeder	B2B Spy-tool
_clsk, _clk	Microsoft Clarity	User Experience Tool
__lotl, __lo_v, __lorid, __lo_uid	Lucky Orange	User Experience Tool

2.2.3 Browser-Verwaltung der vorhandenen Cookies

Viele Internetbrowser bieten dem Nutzer eine ausführliche Darstellung der im Browserspeicher enthaltenen Cookies. Abbildung 2.2 zeigt hierbei diese Darstellung im Google-Chrome-Browser.

2 Grundlagen und Hintergrund

The screenshot shows the 'Alle Websites' tab in the Google Chrome cookie settings. At the top, it displays 'Verwendung' (Usage) and '29,1 MB'. A search bar and a 'Daten löschen' (Delete data) button are also present. Below this, a table lists cookies from various websites, sorted by 'Meistbesucht' (Most visited). Each row shows the website, file size, and number of cookies. To the right of the table is a 'Berechtigungen' (Permissions) section with a 'Berechtigungen zurücksetzen' (Reset permissions) button. This section lists various permissions with dropdown menus for setting them to 'Nachfragen (Standard)', 'Zulassen (Standard)', or 'Blockieren (Standard)'.

Website	Größe	Cookies	Berechtigungen
youtube.com	29,1 MB	16 Cookies	Standort Kamera Mikrofon Bewegungssensoren Benachrichtigungen JavaScript Bilder Pop-ups und Weiterleitungen Werbung Blockieren, wenn Website aufdringliche oder irreführende Werbung anzeigt Hintergrundsynchroneisierung Ton Auto-Downloads MIDI-Geräte USB-Geräte Serielle Schnittstellen Dateien bearbeiten HID-Geräte Geschützte Inhalts-ID's
tiktok.com	3,6 MB	26 Cookies	Nachfragen (Standard)
overleaf.com	1.924 B	6 Cookies	Nachfragen (Standard)
instagram.com	1.959 KB	10 Cookies	Zulassen (Standard)
google.de	4,1 KB	18 Cookies	Zulassen (Standard)
google.com	45,2 KB	37 Cookies	Zulassen (Standard)
etsy.com	29,5 KB	23 Cookies	Blockieren (Standard)
trello.com	1.232 KB	14 Cookies	Zulassen (Standard)
canva.com	14,2 KB	29 Cookies	Automatisch (Standard)
shopify.com	2.739 B	32 Cookies	Nachfragen (Standard)
waterdrop.de	125 KB	34 Cookies	Nachfragen (Standard)
zoom.us	201 B	8 Cookies	Nachfragen (Standard)
heise.de	1.772 B	14 Cookies	Nachfragen (Standard)
aliexpress.com	782 KB	20 Cookies	Nachfragen (Standard)

(a) Cookie-Auflistung

(b) Cookie-Berechtigungen

Abbildung 2.2: Google Chrome Cookie-Verwaltung

Die hier enthaltenen Cookies werden kategorisiert aufgelistet. Hierbei können diese (einzelnen) gelöscht werden. Zudem besteht zusätzlich die Möglichkeit, einzelne Berechtigungen (welche normalerweise innerhalb der Cookies gespeichert werden) bezüglich verschiedener Webbrower-/PC-Elemente, manuell zu bearbeiten.

2.3 Shop-Systeme

Grundlegend ist ein Shop-System eine einfache Software, mit welcher ein Online-Shop von einem Unternehmen bzw. von einem Händler betrieben werden kann. Anders als bei externen Plattformen (Amazon, eBay usw.) befinden sich Unternehmen mit einem eigenen Online-Shop auf ihrer eigenen Plattform, was diesen wiederum zusätzliche Vorteile ermöglicht. Unternehmen welche planen, ihre Produkte und Dienstleistungen Online zu verkaufen, können hierbei auf das Errichten einer eigenen Online-Shop-Infrastruktur mittels eines eigenen Softwareentwickler-Teams verzichten. Vorhandene Ressourcen können direkt in den Aufbau des eigentlichen Online-Shops investiert werden, dies spart unter anderem erhebliche Mengen an Zeit und Geld.

2 Grundlagen und Hintergrund

Shop-Systeme unterscheiden sich hauptsächlich in ihrer „Lösungsart“. Hierbei wird zwischen sogenannten Software-as-a-Service (SaaS) sowie Platform-as-a-Service (PaaS) Lösungen unterschieden. Zusätzlich existieren verschiedene Open-Source Lösungen, welche die beiden vorherigen Lösungen mit zusätzlichen Funktionen und Vorgehensweisen ergänzen. Jedes Unternehmen muss hierbei für sich selbst entscheiden, welche Lösungsform für die eigene Unternehmensstruktur geeignet ist. Neben der allgemeinen Unternehmensgröße sowie den finanziellen Mitteln, spielen auch der vorhandene zeitliche Aspekt eine größere Rolle. Die Nutzung einer PaaS-Lösung wird mehr Zeit für die Entwicklung einer Online-Shop-Präsenz in Anspruch nehmen als eine SaaS-Lösung.

Generell unterscheiden sich SaaS und PaaS hierbei in der Bereitstellung der verschiedenen Module für einen Online-Shop. Während Shop-Systeme, welche mit der SaaS-Lösung arbeiten ein „Komplett Paket“ zur Verfügung stellen, muss ein Unternehmen, welches sich auf eine PaaS-Lösung verlässt, deutlich mehr Ressourcen in die Entwicklung eines Online-Shops investieren. Bei letzterem werden nur die äußerlichen Grundlagen wie Betriebssysteme und Datenbanken zur Verfügung gestellt, eigene Software und Anwendungen für den Online-Shop müssen vom Unternehmen bzw. vom Händler gestellt bzw. entwickelt sowie gewartet werden.(Rani, 2014)

Im Folgenden werden die gängigsten Shop-Systeme genannt, näher erläutert und miteinander verglichen.

Shopify

Shopify als SaaS-Lösung bietet Unternehmen jeglicher Größe einen nutzerfreundlichen „Shop-Baukasten“, mit welchem per Drag-and-Drop-Editor ein individueller Online-Shop erstellt werden kann. Der Service wird hierbei monatlich über verschiedene Pläne bezahlt, welche je nach Bedarf hinzugebucht werden können. Über den „Shop-Baukasten“ lässt sich hierbei zuverlässig ein eigener Online-Shop erstellen, welcher den gängigsten Normen der Nutzerfreundlichkeit entspricht. Entsprechende „Themes“ als Shop-Oberfläche können bei diesem Service sowohl kostenlos als auch kostenpflichtig erworben werden. Shopify kümmert sich neben der Visualisierung des Shops zusätzlich auch um die verschiedenen Zahlvorgänge und die allgemeinen Transaktionen. Bestellungen werden zuverlässig dokumentiert, die entsprechenden Steuersätze- und Werte können vom Händler praktisch dokumentiert und eingesehen werden. Prinzipiell übernimmt Shopify jegliche Aufgaben und Pflichten, welche bei einer Online-Shop-Präsenz anfallen.

WooCommerce

Eine mögliche Alternative ist das Open-Source-Plugin WooCommerce. Dieses WordPress (Content-Management-System) Plugin ist genau wie WordPress selbst Open-Source und damit auch kostenlos. Generell ist dieses Shop-System für jede Unternehmensgröße geeignet und bietet sehr ähnliche Funktionen wie Shopify. Ein Unternehmen bzw. ein Händler wird auch mit WooCommerce in der Lage sein, einen eigenen Online-Shop mit moderatem Aufwand an Ressourcen zu erstellen.

Shopware

Shopware als SaaS und Open-Source-Lösung ist sehr ähnlich wie der Shopify Service aufgebaut und richtet sich ebenfalls an Unternehmen jeglicher Größe. Der größte Unterschied hierbei liegt allerdings in der „API-first“ Denkweise des Unternehmens. Die entsprechenden Shopware-APIs können sehr leicht in eine bestehende Online-Shop-Infrastruktur eingebunden werden, was eine höhere Flexibilität für Unternehmen bzw. Händler möglich macht. Shopware bietet zudem fertige Business-Modelle für Business-to-consumer (B2C), Business-to-Business (B2B) als auch Direct-to-Consumer (D2C)-Lösungen (Untervariante von B2C). Shopware selbst bietet neuen Unternehmen eine kostenlose Starter Edition an. Für zusätzliche Funktionen muss allerdings eine kostenpflichtige Variante erworben und monatlich bezahlt werden.(shopware, 2022)

Spryker

Der deutsche Anbieter Spryker richtet sich mit seiner PaaS-Lösung an größere Unternehmen, welche für ihre Online-Shop-Präsenz eine deutlich ausgeprägtere Flexibilität benötigen. Die Cloudbasierte PaaS-Lösung richtet sich zudem an B2B- und B2C Business-Modelle. Die versprochene hohe Flexibilität kann durch über 800 API-basierte Module garantiert werden. Hierdurch fallen die monatlichen/jährlichen Gesamtkosten deutlich stärker auf. Preise werden auf individuelle Nachfrage mitgeteilt.(SprykerSystems, 2022)

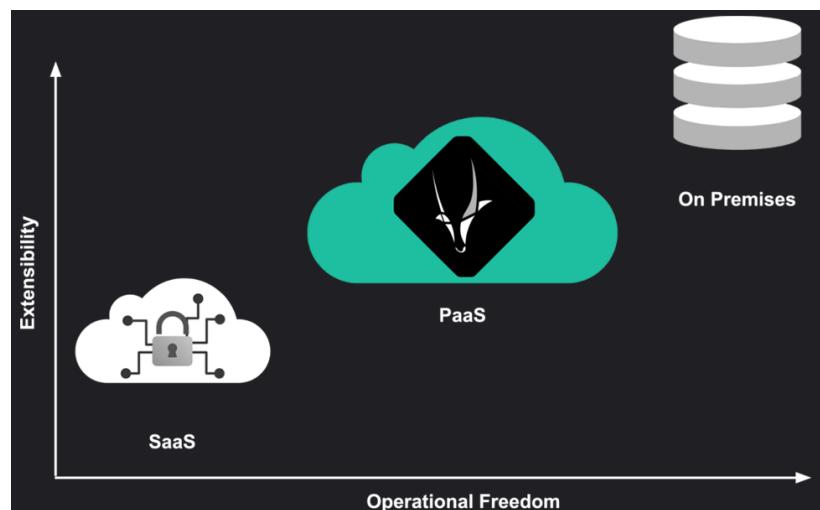


Abbildung 2.3: SaaS vs. PaaS vs. On Premises (spryker.com, 2022)

Abbildung 2.3 zeigt die drei Stufen der vorher genannten Lösungsvarianten. Während SaaS eine moderate Erweiterbarkeit und eine daraus resultierende Handlungsfreiheit besitzt, kann die PaaS Lösungsvariante mit einer deutlichen Steigerung überzeugen. Die nächste Stufe „On Premises“ beschreibt die nächst-höhere Stufe zu PaaS. Hierbei besitzt ein Händler eine Software/Anwendung in einem eigenen (oder auch externen) Rechenzentrum. Diese muss er selbst verwalten und auch regelmäßig warten.

Die Software/Anwendung hat der Händler hierbei von einer externen Partei erworben (bspw. über eine Lizenz), wichtig hierbei ist, dass sich diese Software keinesfalls auf Hardware von dieser externen Partei befinden darf, um diesen Punkt zu erfüllen. Abbildung 2.3 verdeutlicht, wie flexibel PaaS mit einem „gesunden“ Mittelmaß sein kann. Nichtsdestotrotz verbleibt SaaS als eine einfache Lösungsvariante, welche in vielen Fällen absolut ausreichend sein kann.

2.4 Shopify

Shopify ist eine weltweit agierende E-Commerce software-Plattform, mit welcher ein beliebiger Online-Shop erstellt werden kann. Händler bzw. Unternehmen können über einen einfachen „Drag & Drop“ Editor einen beliebigen individuellen Online-Shop erstellen, mit welchem sie online ihre Waren vermarkten können. Zusatzfunktionen wie Zahlungen, Marketing, Versand, Rechtssicherheit, Domain sowie selbstverständlich auch Produkte können einfach über die vorhandene Shopify Oberfläche bearbeitet bzw. ergänzt werden. Über Shopify lässt sich ein vollkommen funktionsfähiger Online-Shop erstellen, hierbei kommt die Software erst an ihre Grenzen, sobald eigene Ideen in Form von Code umgesetzt werden soll, da in diesem Fall ein Softwareentwickler den vorhandenen Code des erworbenen Themes bearbeiten muss. Für viele Probleme lässt sich allerdings auch eine passende App in das vorhandene Theme integrieren, diese Apps können kostenlos, aber auch kostenpflichtig über den vorhandenen Shopify-Store erworben werden (siehe Kapitel 2.4.1).

Shopify selbst beinhaltet eine 14-Tägige Testversion, welche danach in einer bezahlten Version weitergeführt werden kann. Im Jahr 2021 wurden 1,7 Millionen Online-Shops über Shopify in 175 Ländern erstellt (shopify, 2022), (Ahmed, 2022). Die Gesamt-Umsatzzahlen sind hierbei rasant angestiegen. Während im Jahr 2019 noch insgesamt 61 Milliarden USD umgesetzt wurden, waren es im Jahr 2020 bereits 2.929 Milliarden USD (NYSE:SHOP, 2022a) (NYSE:SHOP, 2022b). Allgemein besitzt Shopify in den USA einen Marktanteil von 8,6% der vorhandenen Online-Shops. 32% der Online-Händler verwenden Shopify als favorisierte Shop-System. Zusammenfassend wird Shopify ein deutliches zukünftiges Wachstum vorausgesagt (Rancea, 2022).

2.4.1 Shopify-App-Store

Der Shopify-App-Store ist eine von Shopify selbst zur Verfügung gestellte API-Anbindung, welche 2009 in die Software integriert wurde (Duncan, 2022). Die API-Anbindung erlaubt es externen Entwicklern, eigene Anwendungen (Apps) für Online-Händler zu entwickeln, welche wiederum diese Anwendungen in ihre Online-Stores integrieren können. Entwickler können sich hierbei auf vorhandene Probleme der Händler, bzw. neue Funktionalitäten konzentrieren. Beispielsweise kann ein Händler eine entwickelte App in dessen Online-Shop integrieren, welche einen Cookie-Banner am unteren Bildschirmrand anzeigt und den aktuellen Nutzer über die Verwendung von Cookies aufklärt. Im Jahr 2021 konnten über 7.000 Apps im Shopify-App-Store erfasst werden (Rancea, 2022).

2.4.2 Externe Anbindungen

Neben den verschiedenen Shopify-App-Store-Entwicklern hat zusätzlich der Shop Betreiber (Online-Händler bzw. Unternehmen) die Möglichkeit, auf diese Shopify-API zuzugreifen. Hierdurch können beispielsweise Unternehmens-interne Anwendungen implementiert und an den vorhandenen Online-Shop angebunden werden. Zusätzlich besteht die Möglichkeit, bereits bestehende Unternehmensinterne Strukturen an den vorhandenen Online-Shop anzubinden, um so den Online-Shop über die Unternehmenslandschaft und hierdurch zusätzlichen Funktionen zu erweitern.

Ein Online-Shop-Betreiber könnte hier beispielsweise einen eigenen Cookie-Banner entwickeln, um diesen als Anwendung (App) in den vorhandenen Online-Shop zu integrieren. Dieses Szenario trifft beispielsweise dann zu, wenn ein Unternehmen nicht mit den im Shopify-App-Store angebotenen Cookie-Banner-Apps zufrieden ist, bzw. diese nicht die Erwartungen des Unternehmens erfüllen.

2.4.3 Identifizierung und Erfassung von Shopify-Shops

Ein Online-Shop, welcher Shopify als Grundlage verwendet, kann über unterschiedliche Wege erkannt werden. Der visuell schnellste Weg ist das einfache Erkennen der bekanntesten Shopify Themes. Das wohl bekannteste und auch meist genutzte Theme ist das so genannte „Debut“ Theme. Dieses Standard-Theme wird kostenlos in jeden neu erstellten Online-Shop geladen und erfüllt sämtliche Anforderungen der modernen Nutzerfreundlichkeit. Auch wenn viele Online-Händler dieses Theme verwenden, kann hierdurch nicht jeder Shopify-Shop erfasst werden.

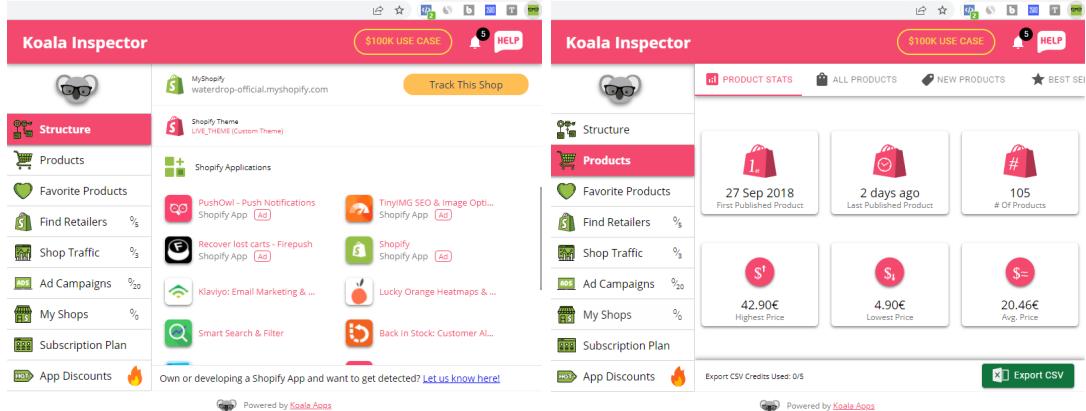
Eine weitere Möglichkeit ist die Erfassung des Wasserzeichens „powered by Shopify“ am untersten Punkt eines Themes. Hierdurch kann theoretisch ein Shopify-Shop eindeutig als dieser erfasst werden. Das Problem an dieser Stelle ist allerdings, dass viele Online-Händler diesen Zusatz über die entsprechende interne HTML-Datei entfernen, um eine gewisse zusätzliche Seriosität zu erlangen. Dieser Vorgang kann in wenigen Minuten und mit Hilfe einer kurzen Suche im Internet von jedem Online-Händler durchgeführt werden, wodurch diese Methode zur Erfassung von Shopify-Shops ebenfalls nicht mehr vollständig zuverlässig ist.

Neben diesen beiden äußerlichen Möglichkeiten besteht die Option der Nutzung einer speziell für dieses Problem entwickelten Browser-Erweiterung. Diese kann in die gängigsten Browser integriert werden, um so über einen einfachen Klick auf diese Erweiterung herauszufinden, ob der aktuelle Shop auf der Shopify-Software basiert oder nicht. Eine bekannte Erweiterung ist hierbei der „Koala Inspector“, welcher nicht nur anzeigt, ob der aktuelle Shop ein Shopify-Shop ist, sondern auch zusätzliche Informationen wie z.B Datum der Shop-Eröffnung, neuestes Produkt, Bestseller Produkt, Anzahl der Produkte, integrierte Apps usw.).

2 Grundlagen und Hintergrund

Auch wenn diese Methode im Gegensatz zu den beiden vorherigen Methoden deutlich zuverlässiger erscheint, besitzt auch diese Herangehensweise ihre Grenzen, da hiermit immer nur der aktuelle Shop betrachtet werden kann. Die beschriebene Browser-Erweiterung muss hier bei jedem Online-Shop manuell aufgerufen werden.

Abbildung 2.4 zeigt den allgemeinen Aufbau der „Koala Inspector“ Browser-Erweiterung. Hierbei werden die informationsreichsten Kategorien der Erweiterung präsentiert.



(a) Allgemeine Informationen des Shopify Online-Shops (b) Produktinformationen des Shopify Online-Shops

Abbildung 2.4: Die Browser-Erweiterung „Koala Inspector“

Die letzte Methode ist das einfache Abfragen der verschiedenen Hosting Informationen über eine gegebene IP-range. Shopify selbst nennt hier in offiziellen Dokumentationen (Shopify, 2022) die IP-Adressen:

- 23.227.38.32
- 23.227.38.65

Über ein entsprechendes Online-IP-Informations-Tool (bspw. myip.ms) lassen sich hieraus verschiedene Informationen zu den übergebenen IP-Adressen extrahieren.

Abbildung 2.5 zeigt eine Datenbanksuche zu den vorher genannten IP-Adressen. In diesem Fall ist aus den beiden IP-Adressen eine range gebildet worden, in welcher gesucht werden soll. Abbildung 2.5 zeigt auch, dass die vorhandenen IP-Adressen theoretisch nicht zwingend übergeben werden müssen, da myip.ms auch nach dem IP-Owner „Shopify, Inc“ suchen kann. Eine zusätzliche Filterung kann über den Webseiten-Namen vorgenommen werden, hierbei können beispielsweise alle deutschen Seiten extrahiert werden, indem man nach dem Merkmal „.de“ sucht. Bei einem Klick auf „Search“ werden alle offiziellen Shopify-Stores mit einer „.de“-Domain geladen und tabellarisch angezeigt. Einzelne Webseiten-Informationen können nun angezeigt, analysiert und verglichen werden.

Abbildung 2.5: Shopify IP-Owner/IP-Bereich-Suche auf myip.ms

2.5 Reguläre Ausdrücke

Mit Hilfe der regulären Ausdrücke (kurz Regex) lassen sich über eine Reihe an definierten Zeichen verschiedene Muster in vorgegebenen Strings/Texten identifizieren. Das Prinzip der regulären Ausdrücke eignet sich daher sehr gut für die Identifizierung von gesuchtem Text in gegebenen Texten. Heutige Text-Editoren verwenden daher dieses Prinzip bei integrierten „Suchen & Ersetzen“-Funktionalitäten, um gesuchten Text zu finden und entsprechend durch neuen Text zu ersetzen. Nachfolgend einige Beispiele.

Das einfache Regex-Muster „abc“ beschreibt eine exakte Übereinstimmung der identifizierten Elemente. Sätze wie „Kennst du das abc?“ sowie „Das müssen wir noch abchecken.“ sind hierbei passende Beispiele für eine Übereinstimmung. Durch das hinzufügen des Metazeichens „*“ (auch „Asteriks“ genannt) im mittleren Teil „ab*c“, findet auch eine Übereinstimmung bei Zeichenketten wie „abbbbc“ und „cbbabbcba“ statt, da dieses für eine beliebige Wiederholung des mittleren Zeichens „b“ steht. (Ionos, 2019)

2.6 Selenium

Selenium ist eine seit 2004 (Selenium, 2022) bestehende Open-Source Software, zum automatisierten Testen von Anwendungen in gängigen Web-Browsern. Selenium besteht dabei aus verschiedenen „Unterkomponenten“ (in die gängigsten IDEs als Erweiterung integrierbar), welche in verschiedensten Situationen eingesetzt werden können (bspw. manuelle „Testaufnahme“, Implementierung, Serverseitiges Webtesten). Das am häufigsten eingesetzte Werkzeug ist allerdings der so genannte Selenium Webdriver (näher behandelt in Kapitel 2.7). Der Webdriver als virtuelle Browser-Kopie nimmt hierbei Eingaben bzw. Befehle durch ein entwickeltes Skript (am häufigsten in Java oder Python implementiert) entgegen und sendet diese Eingaben an den auf dem Rechner befindenden tatsächlichen Browser. Dieser tatsächliche Browser sendet die resultierenden Ergebnisse an den Webdriver zurück, welcher diese wiederum visualisiert.

Diese entwickelten Skripte beschreiben hierbei eine chronologische Abfolge von Anweisungen, welche in einem Test nacheinander abgearbeitet werden sollen.

Allgemein erfolgt der Zugriff über den vorhandenen HTML-Code, welcher vom Webdriver erfasst und vom Skript-Entwickler genutzt werden kann. Der Webdriver kann beispielsweise ein vom Entwickler deklariertes Web-Element erfassen und mit diesem interagieren (bspw. anklicken oder eine Eingabe tätigen). Grundsätzlich ist der Webdriver in der Lage, wie ein realer Nutzer in einem Browser zu agieren, die gängigsten Interaktionen befinden sich hierbei schon in der vorhandenen Selenium Webdriver Bibliothek. Zusätzliche „besondere“ Interaktionen, welche sich nicht in der Standard-Bibliothek vom Selenium Webdriver befinden, können individuell (im besten Fall in der Entwicklungssprache Python) implementiert und als zusätzliches Skript während der Ausführung in das eigentliche Selenium Skript eingebunden werden.

An diesem Punkt muss allerdings erwähnt werden, dass diese Fälle äußerst selten auftreten, da die Standard Selenium-Bibliothek die gängigsten Interaktions-Methoden beinhaltet. Abbildung 2.6 zeigt eine einfache beispielhafte Interaktion mit verschiedenen Web-Elementen in einem Login-Szenario.

```
driver = webdriver.Chrome(executable_path=r"C:\Users\Marvin\Desktop\chromedriver.exe")

# Open website
driver.get('https://stackoverflow.com/')
driver.maximize_window()
time.sleep(3)

# Login
driver.find_element_by_xpath('/html/body/header/div/ol[2]/li[2]/a[1]').click()
driver.implicitly_wait(10)
time.sleep(1)
driver.find_element_by_xpath('//*[@id="email"]').send_keys("EMAIL")
time.sleep(1)
driver.find_element_by_xpath('//*[@id="password"]').send_keys("PASSWORD")
time.sleep(1)
driver.find_element_by_xpath('//*[@id="submit-button"]').click()
driver.implicitly_wait(10)
```

Abbildung 2.6: Selenium Webdriver - Beispielhafte Interaktion mit Web-Elementen

2.7 Webdriver

Wie bereits in dem vorherigen Unterkapitel erwähnt, wird für das Testen von Web-Anwendungen mit Selenium ein so genannter Webdriver benötigt. Dieser kann über die offizielle Selenium Homepage heruntergeladen werden und sollte dabei die gleiche Version wie der sich auf dem Rechner befindende Web-Browser haben. Für die eigentliche Nutzung eines implementierten Skripts wird ein Verweis auf diesen Webdriver benötigt. Da unterschiedliche Web-Browser (Google Chrome, Safari, Mozilla Firefox) existieren, sollte sich die richtige Webdriver Variante auf dem Rechner befinden. Der Selenium Webdriver unterstützt hierbei eine Vielzahl an Programmiersprachen (JavaScript, Ruby, Java, Kotlin, Python usw.). Die am häufigsten verwendeten sind allerdings, wie bereits im vorherigen Kapitel erwähnt, Python und Java.

Eine weitere Besonderheit des Webdrivers ist neben der Erfassung der HTML-Daten die Erfassung der vorhandenen Cookies, welche während der Ausführung der getesteten Webseiten geladen werden. Der Webdriver ist in der Lage, diese zu jedem beliebigen Zeitpunkt auszulesen, zu bearbeiten und zu entfernen.

2.8 Python als Grundlage

Prinzipiell kann der Selenium Webdriver mit jeder kompatiblen Programmiersprache einwandfrei verwendet werden. Der große Vorteil bei der Nutzung von Python in Kombination mit dem Selenium Webdriver ist die Python-typische Reduktion des zu schreibenden Codes, was zu allgemein lesbareren Ausführungs-Skripten führt. Diese verbesserte Lesbarkeit wird nicht nur durch eine Reduktion des vorhandenen Codes garantiert, sondern auch durch eine deutliche „Vereinfachung“ des Codes. Entwickler, welche zuvor sehr wenig mit Python sowie dem Selenium Webdriver gearbeitet haben, können diese Skripte, welche auf diesen beiden Modulen basieren wahrscheinlich besser intuitiv verstehen.

Python kann zudem mit verschiedenen Leistungsstarken IDEs (PyCharm, Spyder, Visual Studio Code etc.) verwendet werden, was der allgemeinen Funktionsweise des Selenium Webdrivers zugutekommt.

3 Zusammenhängende Arbeiten

Das allgemeine Thema des Datenschutzes (besonders im Internet) ist in den vergangenen Jahren verstärkt in den Vordergrund getreten. Durch die sich ausweitende DSGVO ist dieser Prozess dabei deutlich beschleunigt worden. Eine Adaption in der Wissenschaft hat allerdings nur minimal stattgefunden. Es existieren nur wenige wissenschaftliche Veröffentlichungen, basierend auf einem entwickelten Mechanismus, welcher die unterschiedlichen Cookie(-Banner)-Aktivitäten einer Online-Präsenz überwacht und bewertet.

Arbeiten wie (Célestin Matte, 2019) und (Bollinger Dino, 2022) beschreiben hierbei beispielhafte Ansätze dieser wissenschaftlichen Veröffentlichungen. Diese Arbeiten basieren allesamt auf einem entwickelten (oder einem zu entwickelnden) Werkzeug für die Analyse der GDPR-Konformität von Online-Präsenzen. Die Anwendung dieser Werkzeuge sowie die daraus resultierenden Ergebnisse variieren hierbei deutlich.

Beispielsweise werden in (Célestin Matte, 2019) **22.949 europäische Webseiten** mit Hilfe von (halb) automatischen Mechanismen analysiert und bewertet. Hierbei sind die folgenden Informationen generiert worden

- **141 Webseiten** registrieren eine positive Einwilligung, obwohl der Nutzer noch keine Entscheidung bezüglich des Cookie-Banners getroffen hat.
- **236 Webseiten** drängen den Nutzer dazu, die Einwilligung des Cookie-Banners zu akzeptieren, indem die verschiedenen Auswahlmechanismen vorausgewählt sind.
- **27 Webseiten** speichern eine positive Einwilligung, obwohl der Nutzer sich explizit dagegen entschieden hat.

Zudem sind verschiedene „präzisere“ Tests durchgeführt worden, welche die exakte DSGVO-Konformität beweisen sollen. Bei einer Analyse von **560 Online-Präsenzen** konnten bei **54%** dieser, mindestens ein Verstoß gegen die verschiedenen GDPR-Regulierungen erfasst werden.

Die hier aufgelisteten Resultate weisen darauf hin, dass die hier eingesetzten Mittel zu einer Vielzahl an ungenauen Endresultaten geführt haben. Die verwendeten Datensets basieren zudem auf einer „willkürlich gewählten“ zu großen Auswahl (ohne im Vorhinein eine kleinere Menge zu analysieren) an europäischen Domains.

3 Zusammenhängende Arbeiten

Die in dieser Arbeit zu entwickelnde Anwendung soll diese Punkte umgehen, indem das verwendete Domain-Datenset aus Domains einer bestimmten Kategorie (deutsche Shopify-Online-Shops) besteht. Zudem sollen verschiedene Tests bezüglich der Erkennungspräzision im Vorhinein durchgeführt werden, diese sollen präzisere Resultate bei einem größeren Datenset liefern.

4 Entwurf und Implementierung

Das Folgende Kapitel befasst sich mit dem allgemeinen Aufbau der zu entwickelnden Anwendung. Neben einem grundlegenden Entwurf in Form von verschiedenen Diagrammen, welche die allgemeine Funktionsweise der Anwendung beschreiben, soll die darauffolgende Implementierung anhand verschiedener Unterkapitel dargestellt werden. Diese umfasst beispielsweise Themen wie die Identifizierung von Cookies und Cookie-Bannern, das allgemeine Datei-System zum Speichern der Resultate, die Identifizierung von Metadaten sowie die allgemeine Simulation der Nutzer-Interaktion als auch die Generierung, Analyse und Integrierung von verschiedenen Online-Shop-Domains.

4.1 Funktionsweise der Anwendung

Im Folgenden wird der Gesamtprozess der zu entwickelnden Anwendung zur Verdeutlichung der allgemeinen Funktionsweise visualisiert und erläutert. Der Gesamtprozess wird hierbei für ein besseres Verständnis in zwei verschiedene Teilprozesse unterteilt und dabei einzeln erläutert. Die Anwendung in Form der beiden Folgenden Skripte wird hierbei mit Hilfe der Programmiersprache Python entwickelt.

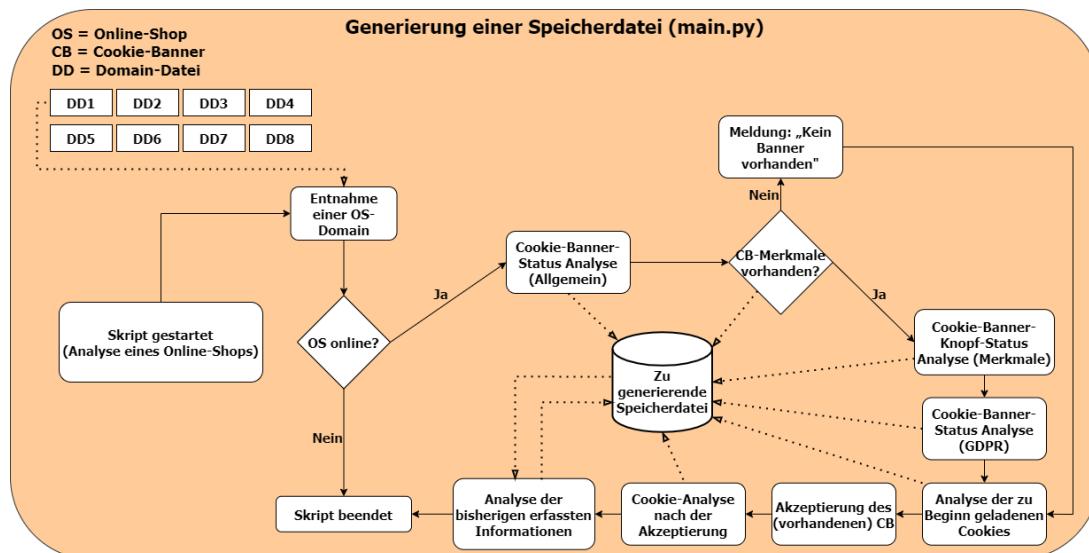


Abbildung 4.1: Beispielhafte Generierung einer Speicherdatei über das main.py-Skript

Die Generierung einer einzelnen Speicherdatei (also die Analyse eines Online-Shops) kann durch Abbildung 4.1 (main.py-Skript) eingesehen werden.

4 Entwurf und Implementierung

Nach dem Start des main.py-Skripts erfasst die Anwendung eine entsprechende Online-Shop-Domain. Hierbei existiert bereits eine Menge an vordefinierten Domain-Dateien, welche wiederum verschiedene Online-Shop-Domains beinhalten. Sobald eine entsprechende Domain erfasst ist, wird überprüft, ob der Online-Shop, welcher durch diese Domain aufrufbar ist, in diesem Moment verfügbar und online ist. Sobald dieser Fall nicht eintrifft, wird die Analyse dieses Online-Shops beendet, wodurch die nächste Domain aus einer Domain-Datei extrahiert wird. Da in diesem Szenario allerdings nur eine Speicherdatei generiert wird, führt diese Aktion zur Beendigung des main.py-Skripts.

Bei einem verfügbaren Online-Shop hingegen werden im nächsten Schritt verschiedene Merkmale bezüglich Cookies sowie Cookie-Banner für spätere Kalkulationen/Entscheidungen erfasst und in die Speicherdatei geschrieben. Zusätzlich wird überprüft, ob anhand dieser Cookie-Banner-Merkmale entsprechende Knöpfe (Annehmen/Ablehnen usw.) vorhanden sind. Die erfassten Informationen werden dabei ebenfalls in die Speicherdatei geschrieben. Basierend auf dem vorhandenen Cookie-Banner, wird überprüft, ob dieser DSGVO-konform ist. Zudem wird eine Analyse der bereits vorhandenen Cookies durchgeführt. Nachdem auch die gerade genannten Informationen in der Speicherdatei gespeichert worden sind, wird der entsprechende vorhandene Cookie-Banner bestätigt, eine weitere Cookie-Analyse durchgeführt und die ermittelten Daten erneut in der Speicherdatei gespeichert.

Der wichtigste und letzte aktive Prozess ist die Analyse aller vorher gesammelten Informationen, welche allesamt in die Speicherdatei geschrieben worden sind. Hierbei werden diese Informationen aus der entsprechenden Speicherdatei extrahiert, analysiert und verschiedene Schlussfolgerungen zur DSGVO-Konformität aufgestellt. Die dabei generierten neuen Informationen werden erneut in die Speicherdatei geschrieben. Die Anwendung endet mit diesem Prozess, eine neue Speicherdatei ist entstanden. In einem realen Szenario extrahiert die Anwendung nun eine weitere Online-Shop-Domain, um eine weitere Analyse und somit die Generierung einer neuen Speicherdatei durchzuführen. In diesem Beispiel hingegen endet die Ausführung mit einer einzelnen Online-Shop-Domain.

Die durch das main.py-Skript (4.1) generierten Speicherdateien bilden in diesem Szenario die Grundlage für die fortführenden Berechnungen und Ausgaben. Das ExtractData.py-Skript (4.2) ist für die Extrahierung aller relevanten Informationen der vorhandenen Speicherdateien zuständig. Nach dem Start des Skripts wird eine entsprechende Speicherdatei erfasst und dabei alle relevanten Informationen aus dieser entnommen. Diese Informationen werden entsprechend zusammengefasst, ergänzt, aufbereitet und über eine Konsoleausgabe dargestellt, was zur Beendigung des Skripts führt. Dieses Skript wird grundlegend nur einmalig ausgeführt, da es lediglich für die einmalige Extrahierung aller relevanten Informationen aus allen vorhandenen Speicherdateien zuständig ist.

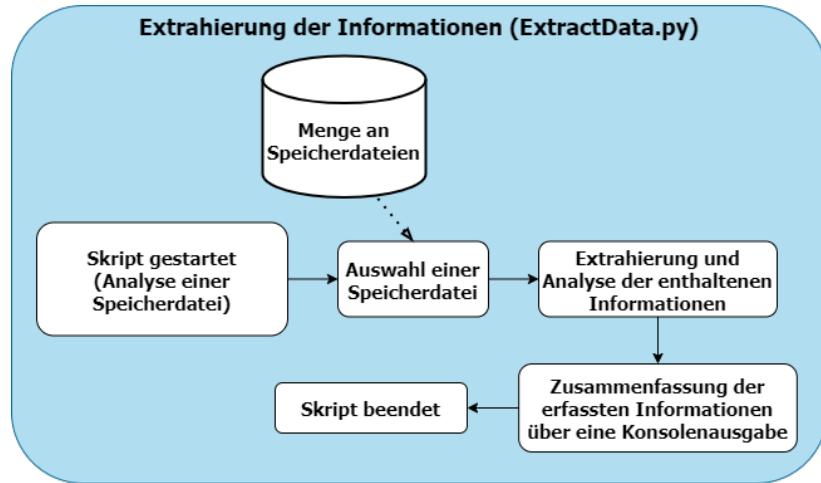


Abbildung 4.2: Beispielhafte Extrahierung und Analyse aller relevanten Informationen einer einzelnen Speicherdatei über das ExtractData.py-Skript

4.2 Banner-Identifizierung

Wie bereits in Kapitel 2.1 erläutert, existieren unterschiedliche Arten von Cookie-Bannern. Während der Entwicklung der Anwendung musste also eine Möglichkeit gefunden werden, wie zwischen diesen Banner-Varianten unterschieden werden kann. Auch wenn sich Shopify-Online-Shops in der grundlegenden HTML-Struktur sehr ähnlich sind, existieren dennoch größere Unterschiede bezüglich des verwendeten Themes sowie bei den verwendeten Cookie-Bannern (falls diese überhaupt im Online-Shop bzw. im HTML-Code zu finden sind). Eine einfache Identifizierung der vorhandenen Cookie-Banner ist in diesem Fall also nicht möglich.

Aus diesem Grund ist ein eigenes System entwickelt worden, welches typische sowie häufig genutzte Wörter für Cookie-Banner erfasst. An diesem Punkt ist auch eine Verwendung von Regulären Ausdrücken für die Erfassung von Wörtern in Betracht gezogen worden. Da Python allerdings eine passende Bibliothek („re“, als Python eigene Bibliothek für Operationen der Regulären Ausdrücke) für dieses Szenario besitzt, ist diese einer eigenen Implementierung der Regulären Ausdrücke vorgezogen worden. Die „re“-Bibliothek wird im späteren Verlauf noch näher erläutert.

Durch das entwickelte System kann initiativ herausgefunden werden, ob überhaupt ein Cookie-Banner vorhanden ist, zu welcher Kategorie dieser Banner gehört (bspw. Standard- oder erweiterter DSGVO-konformer Cookie-Banner) und was für Funktionsweisen dieser Banner besitzt. Für dieses System wird der grundlegende HTML-Code eines zu analysierenden Online-Shops benötigt. Dieser kann durch den entsprechenden Webdriver über die Anweisung „driver.page_source“ ermittelt und als String-Variable im weiteren Programmverlauf genutzt werden.

4 Entwurf und Implementierung

In den folgenden Unterkapiteln wird die allgemeine Verwendung dieser Keywörter in der Anwendung erläutert. Zudem werden alternative Methoden gezeigt, welche während der Entwicklung verwendet worden sind, deren Ergebnisse allerdings nicht optimal waren.

4.2.1 Cookie-Banner Keywörter

Bei der Nutzung von Keywörtern unterscheidet die Anwendung zwischen drei Keywort-Kategorien, diese werden im Folgenden näher erläutert.

GDPR Keywörter

In der Anwendung werden für die Erkennung von DSGVO-konformen Cookie Bannern sowie von allgemeinen Cookie-Bannern diese GDPR Keywörter verwendet. Sobald eine HTML-Datei eines Online-Shops eine bestimmte Anzahl sowie Variation dieser Keywörter enthält, wird dieser Online-Shop als Shop deklariert, welcher einen Cookie-Banner oder sogar einen DSGVO-konformen Cookie-Banner verwendet. Nachfolgend ein paar GDPR Keywörter.

Tabelle 4.1: Beispiele für GDPR Keywörter

„Auswahlmöglichkeiten anpassen“	„Nur Essenzielle Cookies akzeptieren“
„Einstellungen verwalten“	„Cookie-Einstellungen“
„Präferenzen“	„Meine Auswahl bestätigen“

Positive Keywörter

Positive Keywörter werden im Laufe der Berechnungen immer wieder für die allgemeine Erfassung der Cookie-Banner-Funktionalitäten verwendet. Beispielsweise eignen sich diese Keywörter für die Bestätigung der individuellen Cookie-Banner-Knöpfe oder die allgemeine Deklarierung von Online-Shops bezüglich deren DSGVO-Konformität. Beispielhafte Positive Keywörter werden im Folgenden aufgelistet.

Tabelle 4.2: Beispiele für Positive Keywörter

„Alle Akzeptieren“	„Annehmen“
„Alle Cookies zulassen“	„Akzeptieren“
„Fortsetzen“	„Akzeptieren & weiter“
„Verstanden“	„Ich stimme zu“
„ok“	„Einverstanden!“

Negative Keywörter

Die negativen Keywörter sind das passende Gegenstück der positiven Keywörter. Prinzipiell erfüllen sie in der Anwendung die gleichen Aufgaben wie die positiven Keywörter. Der Unterschied hierbei liegt in der Betrachtung der „negativen“ Aspekte von Cookie-Bannern. Beispielsweise wird über diese Keywörter der Status von (nicht) vorhandenen „Ablehnungs“-Knöpfen in Cookie-Bannern ermittelt. Negative Keywörter werden im Folgenden aufgelistet.

Tabelle 4.3: Beispiele für Negative Keywörter

„Ablehnen“	„Alle ablehnen“
„Nur Essenzielle Cookies akzeptieren“	„Nein Danke“
„Nur notwendige Cookies“	„Nur notwendige Cookies Akzeptieren“

In der Anwendung sind die Keywörter in entsprechend benannten Listen manuell aufgelistet. Diese Listen bzw. Keywörter sind hierbei aus einem Datenset an zufällig ausgewählten Online-Shops herausgefiltert worden, sie beschreiben also die erste Version von vorhandenen Keywörtern, welche für die Berechnungen der Anwendung verwendet werden. Eine Option, die entsprechenden Keywörter automatisch zu generieren bzw. zu erfassen, ist aus unterschiedlichen Gründen verworfen worden (mehr dazu in Kapitel 5.7). Der Gesamtprozess der manuellen Keywort-Erfassung und Keywort-Auflistung sowie die andauernde Verbesserung dieser wird in Kapitel 5.1.1 ein weiteres Mal aufgegriffen und erläutert.

Die Keywörter sind ein zentraler Bestandteil der Anwendung, da sie permanent dazu genutzt werden, um einen Cookie-Banner und dessen Funktionalitäten zu erkennen. Aus diesem Grund sollten hierbei keine Fehler während der Ausführung auftreten. Ein wichtiger Zusatz welcher für diese Sicherstellung verwendet wird, ist die bereits erwähnte „re“-Bibliothek. Anstelle eines einfachen String Vergleichs, überprüft diese Bibliothek über die Methode „re.search(keyword, html_source)“, ob ein Keywort genau so in der „html_source“ existiert wie es anfänglich in der Anwendung deklariert worden ist. Bspw. können Situationen, in denen Wörter wie „ok“ mit „look“ gleichgesetzt werden, durch diese Bibliothek verhindert werden, was allgemein dafür sorgt, dass die Anwendung Cookie-Banner-fokussierter arbeitet und generell zuverlässiger ausgeführt wird.

Der Gesamtprozess lässt sich einfacher also so zusammenfassen, dass über die search-Methode der „re“-Bibliothek nach genau einem bestimmten Keywort in einer HTML-Source (welche über den Webdriver erfasst wird) gesucht wird. Sobald dieses in der HTML-Source auffindbar ist (und zusätzliche Filtermethoden überwinden kann), wird dieses Keywort in der Anwendung als „gültig“ eingestuft, wodurch bspw. entschieden wird, ob eine Webseite einen Cookie-Banner bzw. einen DSGVO-konformen Cookie-Banner besitzt oder allgemein auch mit negativen Knöpfen (z.B. „Ablehnen“) arbeitet, also dem Nutzer die Möglichkeit anbietet, den sichtbaren Cookie-Banner abzulehnen.

4.2.2 Screenshot-Funktionalität

Für eine bessere Überwachung der ermittelten späteren Resultate, wird eine einfache Screenshot-Funktionalität in die Anwendung integriert. In fehlerhaften Situationen der Anwendung sowie bei auffälligen Cookie-Bannern soll ein hochauflösender Screenshot des analysierten Online-Shops erstellt und gespeichert werden. Hierdurch wird sichergestellt, dass die Anwendung permanent durch vorliegende „debug“-Screenshots verbessert und erweitert werden kann.

4.2.3 Fehlgeschlagene Methoden

Neben der Keyword-Methode zur Erfassung von Cookie-Bannern sowie den dazu gehörigen Cookie-Banner-Funktionen gibt es drei weitere Methoden, welche diese Aufgaben anfänglich übernehmen sollten. Aus Zeitgründen (durch zu großen Umfang) konnten diese Methoden allerdings nicht umgesetzt werden. Die Methoden werden im Folgenden näher erläutert und spiegeln letztendlich Gedankengänge wider, welche im Laufe der Entwicklung entstanden sind.

Die **erste Methode** ist die Methode der „Struktur-Erfassung“. Hierbei wird ein Muster der verschiedenen Cookie-Banner erkannt und aufgezeichnet. Bei der Analyse eines neuen Online-Shops wird das vorhandene Cookie-Banner-Muster erkannt und mit bisherigen Mustern verglichen. Das Problem bei dieser Methode ist das allgemeine Erfassen der Cookie-Banner-Muster. Da jeder Online-Shop individuell aufgebaut ist und auch individuell entscheidet, wie der eigene Cookie-Banner aussehen und arbeiten soll, ist es schwer, generelle Muster für Cookie-Banner zu erstellen. Zusammengefasst ist diese Methode unzuverlässig und zu ungenau. Aus diesem Grund ist diese Methode verworfen worden.

Bei der **zweiten Methode** wird ein Bild von einem Online-Shop erstellt und einer eigenen Bildverarbeitungssoftware übergeben, welche erkennt, ob sich auf diesem Bild ein Cookie-Banner befindet oder nicht. Zudem soll diese Software erkennen, welche Funktionalitäten dieser Cookie-Banner aufweist (Knöpfe, Einstellungen usw.). Das Problem bei dieser Methode ist der erhebliche Mehraufwand, welcher für eine doch simplere Aufgabe geleistet werden muss. Auch hier ist der Umfang als zu groß eingeschätzt worden, wodurch die Methode ebenfalls verworfen worden ist.

Die **dritte Methode** befasst sich mit der Anbindung externer Open-Source-Projekte, welche nach einiger Recherche gefunden wurden. Hierbei viel allerdings schnell die Entscheidung, dass diese Anbindungen zu unzuverlässig sind und auch allgemein nicht viele (nötige) Möglichkeiten zur Erfassung von Cookie-Bannern und deren Funktionalitäten bieten. Die Ergebnisse gleichen nicht den Aufwand aus, welchen es braucht, diese Projekte an die zu entwickelnde Anwendung anzubinden, wodurch auch diese Methode verworfen worden ist.

Als Beispiel kann hier das „Cookinspect“-Github-Projekt genannt werden (github.com/Perdu, 2021). Dieses verdeutlicht einen Entwurf, welcher durch die zu entwickelnde Anwendung angestrebt wird.

4.3 Cookie-Identifizierung

Wie bereits im Laufe dieser Arbeit erwähnt, ist der Webdriver in der Lage, neben dem vorhandenen HTML-Code einer Webseite, auch die verschiedenen Cookies zu analysieren, welche während der Analyse durch die Webseite geladen werden. Die Cookies können hierbei eingesehen, verändert oder entfernt werden. Für bestimmte Szenarien können auch eigene Cookies in den Ablauf integriert werden.

In der entwickelten Anwendung werden die zu analysierenden Cookies zu verschiedenen Zeitpunkten erfasst, analysiert und dokumentiert. Diese Erfassung kann über die triviale Selenium Webdriver Anweisung „`driver.get_cookies()`“ vorgenommen werden, wobei „`driver`“ der aktuell ausführende Webdriver ist. Durch diese Anweisung wird eine Liste von Cookies generiert, welche durch den aktuellen Gesamtprozess entstanden sind. Im Folgenden werden die beiden Zeitpunkte erläutert, in welchen diese Anweisung für die Cookie-Analyse genutzt wird. Diese Zeitpunkte beziehen sich auf die „Akzeptierung“ des (vorhandenen) Cookie-Banners.

Vor der Cookie-Banner-Akzeptierung

Nachdem der Webdriver einen Shopify Online-Shop geladen hat, wird umgehend der aktuelle Cookie-Status überprüft. Hierbei werden alle Cookies erfasst, welche direkt nach dem Aufbau der Webseite automatisch geladen worden sind.

Nach der Cookie-Banner-Akzeptierung

Sobald der (vorhandene) positive „Akzeptieren“-Knopf des Cookie-Banners betätigt worden ist, wird ein weiteres Mal die soeben genannte Anweisung ausgeführt. Hierbei werden erneut alle aktuellen Cookies des Online-Shops geladen. Darauffolgende Berechnungen vergleichen (unter anderem) diese Ausgabe mit der vorherigen Ausgabe, hierbei entstehen Folgende Informationen:

- Welche Third Party Cookies sind ungenehmigt generiert worden, nachdem die Webseite vollständig geladen wurde?
- Wie viele Cookies sind nach der Akzeptierung hinzugekommen?
- Wurden ungenehmigte Third Party Cookies geladen, obwohl der Nutzer diesen ausdrücklich nicht zugestimmt hat?
- In Kombination mit vorherigen Metriken (Nutzung von entsprechenden positiven-/negativen-/GDPR-Knöpfen): Respektiert der Online-Shop die Entscheidung des Nutzers?
- Woraus in Kombination folgt: Ist der Online-Shop DSGVO-konform?

Die Anwendung dokumentiert jegliche Berechnungen und deren Ergebnisse in Form einer entsprechenden Konsolenausgabe sowie mit Hilfe eines entwickelten Dateisystems, welches für jeden analysierten Webshop eine individuelle Datei erstellt, in welcher alle

relevanten Informationen gespeichert werden. Durch eine entsprechende Metaanalyse der generierten vorhandenen Speicherdateien kann eine Gesamtanalyse für einen „Gesamtüberblick“ durchgeführt werden (in einem späteren Kapitel (ab 5.4) werden die Durchführung einer Metaanalyse sowie deren Ergebnisse näher erläutert).

Nach den jeweiligen Erfassungen der entsprechenden Cookies, sowie allen darauf folgenden Berechnungen, versucht der Webdriver den nächsten Online-Shop zu laden. Hierbei wird zusätzlich in einem Zwischenschritt die Anweisung „`driver.delete_all_cookies()`“ ausgeführt, was die endgültige Löschung der vorhandenen Cookies im Webdriver zur Folge hat. Dieser Prozess kann mit der Löschung der Cookies im eigenen Browser verglichen werden. Hierbei wird sichergestellt, dass bei der Analyse des nächsten Online-Shops keine Cookies des „Vorgänger“-Shops die Berechnungen manipulieren bzw. sabotieren.

4.4 Online-Shop Datei-System

Das bereits im vorherigen Kapitel aufgegriffene Datei-System zur Dokumentierung der generierten Informationen wird im Folgenden näher erläutert. Zudem wird zusätzlich verdeutlicht, warum andere Varianten des Datei-Systems verworfen worden sind.

Das entwickelte Datei-System wird von der Anwendung genau dann aufgerufen, sobald generierte Informationen gespeichert werden sollen. Hierbei wird die Python-eigene „`open()`“-Methode verwendet, welche einen Pfad entgegennimmt, welcher wiederum einen individuellen Namen enthält. Dieser Name ist in diesem Fall der Name des analysierten Online-Shops. Die „`open()`“-Methode überprüft daraufhin, ob bereits eine Datei über diesen Pfad existiert oder ob diese noch erstellt werden muss, um diese anschließend zu verwenden. Die „`open()`“-Methode hinterlässt anschließend eine Variable, mit welcher verschiedene Informationen in die generierte Datei geschrieben werden können. Durch das Aufrufen des Datei-Systems können allgemein also Informationen, welche während einer Analyse eines Online-Shops entstanden sind, in einzelnen individuellen Dateien gespeichert und später wieder abgerufen werden.

Bei der Analyse von einzelnen Online-Shops entstehen hierbei auch nur einige wenige Speicherdateien. Sobald sich die Menge der zu analysierenden Online-Shops allerdings erhöht, vergrößert sich auch die Menge der generierten Speicherdateien. Dieses Prinzip wirkt auf den ersten Blick in die Programmstruktur „erschlagend“, erreicht allerdings die besten Laufzeitergebnisse für diesen Anwendungsfall. Hierbei muss beachtet werden, dass diese generierten Speicherdateien inklusive deren Informationen im späteren Verlauf der Anwendung wieder geöffnet und analysiert werden müssen. Dabei hat sich herausgestellt, dass einzelne Speicherdateien am übersichtlichsten sind und allgemein effizienter (über entsprechenden Programmcode) verarbeitet werden können.

Im Folgenden werden kurz verschiedene Methoden vorgestellt, welche vom aktuellen Datei-System abgelöst worden sind, somit also verworfen wurden.

Bei der **ersten Methode** wird vollständig auf das Datei-System verzichtet, hierbei werden lediglich Konsolenausgaben vorgenommen. Bei einigen wenigen Online-Shop-Analysen mag diese Methode zwar funktionieren, sobald sich die Menge der zu analysierenden Online-Shops allerdings erhöht, kann diese einfache Konsolenausgabe sehr unübersichtlich werden. Außerdem können die Ergebnisse in einem späteren Programmverlauf nicht erneut aus der Konsole extrahiert werden, um diese in weitere Berechnungen einzubeziehen.

Bei der **zweiten Methode** existiert eine Speicherdatei, welche alle generierten Informationen aller analysierten Online-Shops enthält. Hierdurch erhält man zwar eine sauberere Ordnerstruktur, die Speicherdatei selbst allerdings wird sehr unübersichtlich. Zudem erschwert eine einzelne Speicherdatei die spätere Erfassung der generierten Informationen (über den Code) für die durchzuführende Metaanalyse. Bei dieser Methode sind zudem einzelne Probleme beim Schreiben der Informationen in die Speicherdatei aufgetreten, hierbei kam es zu Laufzeitproblemen, was häufig auch die Anwendung abstürzen lassen hat.

Die **dritte Methode** ist eine Erweiterung der **zweiten Methode**. Hierbei existiert nicht nur eine zentrale Speicherdatei, sondern mehrere einzelne Speicherdateien, welche einzelne Kategorien widerspiegeln. Hierbei werden analysierte Online-Shops in diese Kategorien eingeteilt und gespeichert. Der Grund, warum diese Methode allerdings ebenfalls verworfen worden ist, ist der Gleiche wie bei **Methode zwei**. Der benötigte Code, um die Informationen in diesen Schreibdateien zu extrahieren, ist deutlich aufwändiger und komplexer als bei der Methode der vielen einzelnen Speicherdateien.

Generell ist eine einfache Extrahierung und Erfassung der generierten Informationen aus den Speicherdateien präferiert worden. Wenige Speicherdateien erzeugen zwar eine übersichtlichere Projektübersicht, erschweren allerdings die Extrahierung der generierten Informationen für die spätere Metaanalyse. Der zu entwickelnde Code für diese Metaanalyse fällt hierdurch sowohl kürzer als auch effizienter aus.

Anhang [Aufbau einer Speicherdatei] zeigt eine einzelne generierte Speicherdatei. Diese enthält neben verschiedenen Metadaten (wird in den folgenden Kapiteln erläutert), die erfassten Cookies zu den verschiedenen Zeitpunkten (vor/nach der Akzeptierung des Cookie-Banners) sowie eine endgültige Schlussfolgerung, also eine Analyse der DSGVO-Konformität des analysierten Shopify-Online-Shops.

4.5 Identifizierung von Metadaten

Metadaten werden sowohl für die Wiedererkennung von Speicherdateien sowie für die allgemeine Analyse von Online-Shops benötigt. Der Entschluss, ob ein Online-Shop DSGVO-konform ist, basiert neben dem vorhandenen Banner, sowie den dazugehörigen Cookies auch auf diesen Metadaten. Anhang [Metadaten-Auflistung] listet diese Metadaten auf, erläutert sie und beschreibt zudem, welchen Einfluss diese auf die finale Entscheidung bezüglich der DSGVO-Konformität haben.

Zudem wird gezeigt, dass nicht alle Metadaten einen Einfluss auf die Entscheidung haben, ob ein Online-Shop DSGVO-konform ist oder nicht. Viele dienen im späteren Verlauf auch zur Analyse zum allgemeinen Online-Shop-Verhalten bezüglich der (nicht) vorhandenen Cookie-Banner (z.B wie viele Online-Shops einen „normalen“ Cookie-Banner verwenden, dieser allerdings nicht DSGVO-konform ist). Im Laufe dieser Arbeit werden diese Daten/Ergebnisse noch näher vorgestellt.

4.6 Simulation der Nutzer-Interaktion

Wie bereits im Laufe dieser Arbeit erläutert, muss die entwickelte Anwendung die präsentierten Cookie-Banner Akzeptieren, um das allgemeine Cookie-Verhalten zu analysieren und zu dokumentieren. Hierfür muss die Anwendung mit Hilfe des Webdrivers auf Anfrage in der Lage sein, einen angezeigten Cookie-Banner jeglicher Form und Größe zu erfassen und zu bestätigen.

Die Anwendung löst dieses Problem mit Hilfe der Abfragesprache XPath. Mit Hilfe dieser Sprache lassen sich Punkte bzw. Knoten in einem XML-Dokument exakt erfassen und bestimmen. Im Szenario eines Web-Browsers lassen sich über XPath also exakte „Koordinaten“ für ein Web-Element bestimmen.

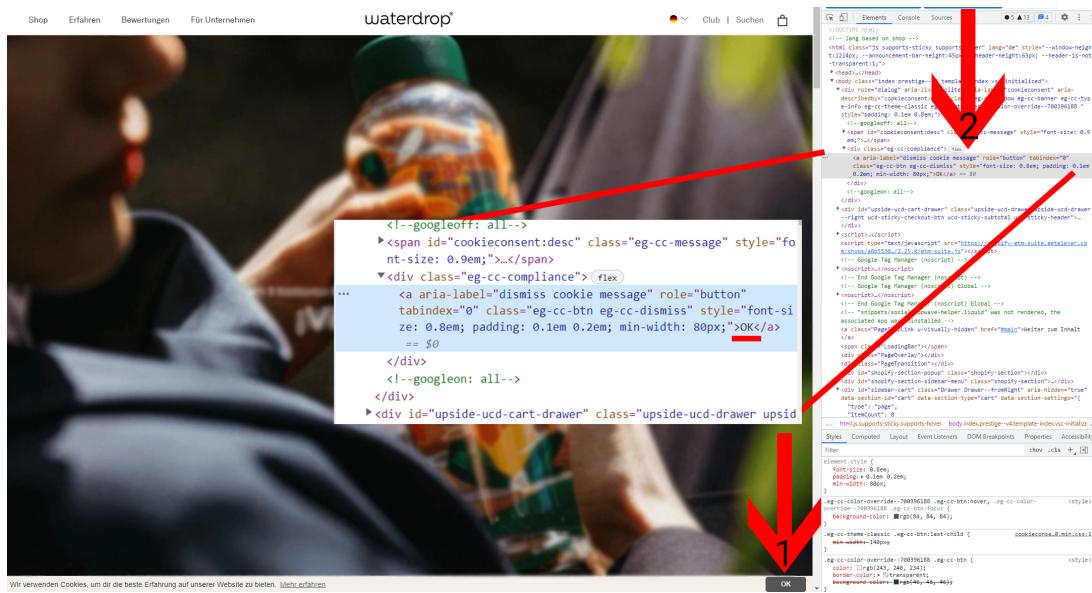


Abbildung 4.3: Identifizierung des „Akzeptieren“-Knopfes in einem Cookie-Banner

Abbildung 4.3 zeigt ein Beispielszenario, in welchem über die eingebaute Entwicklerkonsole des Google Chrome Browsers ein Web-Element (Pfeil 1) über den vorhandenen HTML-Code (Pfeil 2) erfasst wird. Theoretisch kann bei diesem Web-Element über einen einfachen „Rechtsklick“ der entsprechende XPath mit Hilfe des Chrome-Browsers kopiert und verwendet werden.

```
for keyword in positive_cookie_banner_buttons:  
    buttons = driver.find_elements_by_xpath("//*[contains(text(), '" + keyword + "')])")
```

Abbildung 4.4: Erfassung des korrekten XPaths eines gesuchten Web-Elements

Die entwickelte Anwendung allerdings nutzt einen Code-Abschnitt (siehe Abbildung 4.4), welcher die automatische Suche nach entsprechenden (möglichen) gesuchten Web-Elementen erlaubt. Hierbei werden alle positiven Keywörter mit dem „Standard“-XPath kombiniert, welche daraufhin in der vorhandenen HTML-Datei gesucht werden. Die gefundenen Elemente werden schließlich gefiltert und aussortiert, bis ein finales Element (das gesuchte) verbleibt und im späteren Programmverlauf betätigt werden kann.

Im Szenario von Abbildung 4.3 sucht die Anwendung also in der vorhandenen HTML-Datei nach auf positiven Keywörtern basierenden XPath's. Da in diesem Szenario nach einem „ok“-Knopf gesucht wird, gelangt die Anwendung bei dem positiven Keywort „ok“ an mindestens einen möglichen XPath. Sehr häufig werden hier verschiedene XPath's erfasst, da der Webdriver nach allen Web-Elementen sucht, welche ein „ok“ beinhalten. Diese Elemente können neben Knöpfen unter anderem normaler Text oder Bild-Unterschriften/Anmerkungen sein. Mit Hilfe eines entsprechenden Ausfilterungs-Systems werden diese erfassten Elemente wieder entfernt. Der verbleibende XPath spiegelt also das gesuchte Web-Element (in diesem Fall ein „ok“-Knopf) wider, welches über die Webdriver-eigene Methode „.click()“ betätigt werden kann.

Auf diese Weise sucht und erfasst die Anwendung auch nach „negativen“ Knöpfen. Die hier gesuchten XPath's werden über die negativen Keywörter gesucht und über dieselben Filtermethoden gefiltert. Die Anwendung betätigt diese XPaths nicht, schaut allerdings, ob sie „theoretisch“ anklickbar wären. Trifft dies zu, wird der analysierte Shopify-Online-Shop als ein Shop eingestuft, welcher dem Nutzer einen „negativen“ Knopf anbietet (bspw. „Ablehnen“).

4.7 Generierung von benötigten Online-Shop-Domains

Wie bereits in Abbildung 2.5 aus dem Kapitel 2.4.3 gezeigt, können Shopify-Online-Shops aus einer bestimmten Region unproblematisch über eine einfache IP-Suche gefunden und erfasst werden. Zum Testen der entwickelten Anwendung, sowie für die spätere Generierung der verschiedenen Resultate, wird eine ausreichend große Menge an Shopify-Online-Shop-Domains benötigt.

4 Entwurf und Implementierung

Detailed Search

Full/Part Website Name: .de Web Server Location: - Select Country - Site Popular Rating:

Website IP Address: 23.227.38.32 Website IPv6 Address: - 23.227.38.65 IP Owner/Parent IP Owner/Hosting: Shopify, Inc

Top Level Host (Reverse IP): - Select One - DNS Records: - Select One - Popularity (visitors per day):

Search Reset

Download in Excel/CSV Format Report Information

No	Web Site	Website IP Address	Web Hosting Company / IP Owner	Web Hosting / Server IP Location	Web Hosting City	World Site Popular Rating	Diagram
1	saguaroschuh.de	23.227.38.32	Shopify, Inc	Canada		# 28,361	
2	coronapoint.de	23.227.38.65	Shopify, Inc	Canada		# 91,997	
3	happy-posters.de	23.227.38.65	Shopify, Inc	Canada		# 96,930	
4	boombywaleria.de	23.227.38.32	Shopify, Inc	Canada		# 98,802	
5	oneodio.de	23.227.38.65	Shopify, Inc	Canada	Ottawa	# 105,610	
6	rosental.de	23.227.38.32	Shopify, Inc	Canada		# 116,411	
7	deinschuhwerk.de	23.227.38.32	Shopify, Inc	Canada		# 123,693	
8	weinfuerst.de	23.227.38.65	Shopify, Inc	Canada		# 127,478	
9	morenutrition.de	23.227.38.65	Shopify, Inc	Canada		# 131,044	
10	levlup.de	23.227.38.32	Shopify, Inc	Canada		# 137,923	
11	boc24.de	23.227.38.32	Shopify, Inc	Canada		# 155,508	
12	colorfulstandard.de	23.227.38.65	Shopify, Inc	Canada		# 167,230	

1 - 50 of 3,288 records

Abbildung 4.5: Shopify IP-Owner/IP-Bereich-Detailsuche auf myip.ms

Abbildung 4.5 zeigt eine Domain-Abfrage auf myip.ms. Hierbei wird nach deutschen „.de“-Domains gesucht, welche den IP-Owner *Shopify* als Merkmal besitzen, sich also in dem IP-Bereich aufhalten, welcher in Kapitel 2.4.3 bereits genannt ist. Bei der Suche werden insgesamt 3.288 übereinstimmende Shopify-Domains zurückgegeben (Pfeil 1). Zum Testen der Anwendung reichen allerdings 100 „qualitative Domains“ vorerst aus. „Qualitative Domains“ bezeichnen hierbei Domains, welche aus verschiedenen Shop-Qualitäts-Kategorien stammen. Hierbei werden Online-Shops ausgewählt, welche in der deutschen Rangliste „höher“ aufgeführt werden, sowie Online-Shops, welche dieses Kriterium nicht erfüllen.

Durch dieses Vorgehen wird sichergestellt, dass die Anwendung mit verschiedenen Arten von Cookie-Bannern konfrontiert wird. Eine einfache Recherche der 20 am besten gelisteten Online-Shops zeigt eine starke Ähnlichkeit der verwendeten Cookie-Banner. Die Anwendung analysiert hierdurch also immer wieder die gleichen Szenarien, was dem Prozess des Testens schadet und zudem dazu führt, dass die Anwendung sich nur wenig weiterentwickeln kann.

4.8 Analyse der erfassten Online-Shop-Domains

Bei den von myip.ms übergebenen Domains bzw. Online-Shops handelt es sich zum größten Teil um „Nischen-Shops“. Hierbei werden einzelne Käufergruppen bedient, welche sich für ein spezielles Themengebiet interessieren. Hieraus ergibt sich, dass diese Online-Shops „auf den ersten Blick“ unbekannter erscheinen als bspw. die „obigen Top 25“ der Rangliste, bei näherer Betrachtung fallen allerdings einige wenige Online-Shops auf, welche den persönlichen Interessen entsprechen und dadurch „bekannter“ auf ein einzelnes Individuum wirken. Zusammenfassend lässt sich also sagen, dass es sich bei diesen Online-Shops zwar um „weniger bekannte“ Nischen-Shops handelt, diese allerdings Online-Shops widerspiegeln, welche von Menschen alltäglich verwendet werden.

4.9 Integrierung der Online-Shop-Domains in die Anwendung

Damit die ermittelten Domains in die Anwendung transferiert und von dort genutzt werden können, sind diese in eine entsprechende Excel-Tabelle transferiert worden. Die Anwendung kann auf diese Tabelle nicht direkt zugreifen, wodurch diese mit Hilfe der in Excel integrierten Filtermethoden bearbeitet worden ist. Die daraus entstandenen Shop-Domains sind daraufhin in eine entsprechende Textdatei geladen worden. Die entwickelte Anwendung kann aus der generierten Textdatei die einzelnen Zeilen auslesen, die Domains erfassen und diese letztendlich auch nach entsprechenden Formatierungen öffnen.

Anhang [Domain-Text-Datei-Auflistung] zeigt den Inhalt dieser Textdatei. Diese beschreibt also eine beispielhafte Auflistung der ermittelten Domains zum Testen der entwickelten Anwendung.

5 Erzeugung und Evaluation

Anders als bei Kapitel 4, welches sich mit dem allgemeinen Entwurf und der Implementierung der Anwendung befasst hat, basiert dieses Kapitel auf der darauffolgenden Erzeugung von verschiedenen Resultaten bezüglich der DSGVO-Konformität von Shopify-Online-Shops sowie deren Evaluation anhand verschiedener Aktualisierungen des vorhandenen Codes der Anwendung.

Hierfür wird zuerst eine Validierung der Anwendung durchgeführt. Dabei wird überprüft (auch manuell), wie präzise die Anwendung arbeitet und was im Ablauf der Anwendung verbessert werden kann. Zudem werden Berechnungen zu den verschiedenen Berechnungsgeschwindigkeiten aufgestellt, um einen grundlegenden Überblick über die später benötigte Zeit zu erhalten (für die Hauptanalyse).

Daraufhin folgt die eigentliche Hauptanalyse, welche unter anderem die Erzeugung eines größeren Datensets an Online-Shop-Domains sowie die Extraktion aller relevanten Informationen aus den vorhandenen Speicherdateien beinhaltet.

Basierend auf dieser Hauptanalyse werden die verschiedenen Ergebnisse dargestellt und erläutert. Zudem werden Probleme genannt, welche während der Entwicklung und der Erzeugung der Resultate aufgetreten sind.

5.1 Validierung der entwickelten Anwendung

Die Folgenden Unterkapitel beschreiben das Testen der entwickelten Anwendung. Hierbei wird zuerst eine passende Test-Art ausgewählt und daraufhin entschieden, wie viele Tests für dieses Szenario ausreichend sind. Zudem wird die Integrierung der aus den Tests resultierenden Ergebnisse (aktualisierter Programmcode) in die Anwendung sowie dessen Resultate beschrieben. Abschließend wird basierend auf den Test-Resultaten die allgemeine Erkennungspräzision der Anwendung bezüglich der analysierten Online-Shops genannt und verdeutlicht.

5.1.1 Manuelle Überprüfung eines vordefinierten Datensets

Beim Testen der Anwendung soll sichergestellt werden, dass diese genau die Aspekte eines Online-Shops erfasst, welche auch dort abgebildet sind. Hierbei muss ein Abgleich zwischen Anwendung und Online-Shop stattfinden. Die Frage der Ausführung bleibt jedoch unbeantwortet. Sollen die zum Testen ermittelten Shop-Domains nacheinander und vor allem „unbeobachtet“ abgearbeitet werden? Oder wäre eine manuelle Übergabe der einzelnen Domains sowie eine permanente Betrachtung der Anwendung zielbringender? Nach kurzer Überlegung sowie einzelnen Tests hat sich die zweite Variante aus

Sorgfaltsgründen durchgesetzt. Hierbei hat sich herausgestellt, dass eine automatische Durchführung zu einer „zu schnellen“ Ausführung führt, wodurch eine permanente Betrachtung sowie eine Validierung der erzeugten Ergebnisse nicht möglich ist. Bspw. ist es bei der automatischen Variante in wenigen Sekunden nicht möglich zu sehen, ob die Anwendung korrekt einen Online-Shop als DSGVO-konform deklariert oder nicht. Aus diesem Grund wird die Anwendung manuell über einzelne Domains (aus dem in Kapitel 4.7 ermittelten Online-Shop-Domains) getestet.

Eine einzelne Domain wird also manuell in die Anwendung kopiert, diese wird daraufhin gestartet, um anschließend die übergebene Domain zu analysieren. Die Anwendung dokumentiert die generierten Ergebnisse, indem sie diese in eine einzelne Speicherdatei (Erläutert in Kapitel 4.4) schreibt. Diese Ergebnisse werden nun im nächsten Schritt manuell überprüft. Hierbei wird die übergebene Domain ein weiteres Mal manuell geladen, um sich ein generelles Bild über die Lage der DSGVO-Konformität zu machen. Anschließend werden die manuell erfassten Ergebnisse mit den Ergebnissen der Anwendung verglichen und dabei entschieden, ob die Anwendung bezüglich der erfassten Metriken sowie Resultate richtig oder falsch entschieden hat. Dieser Prozess wird für jede der 100 erfassten Test-Domains wiederholt. Die erfassten Test-Resultate (zu ändernder Programmcode) werden dokumentiert und nach dem Test in die Anwendung integriert.

Die ermittelten Test-Ergebnisse werden im späteren Verlauf in Kapitel 5.1.2 zusammengefasst und näher erläutert.

Eine allgemeine Besserung der Test-Resultate wird garantiert, indem die Integrierung der ermittelten Test-Ergebnisse (aktualisierter Programmcode) in die Anwendung sichergestellt wird. Die Integrierung dieser Ergebnisse wird im Folgenden näher erläutert.

Integrierung der Ergebnisse in die bestehende Anwendung

Die bei der Integrierung der generierten Test-Ergebnisse aufgetretenen Folgefehler in der Anwendung werden entsprechend behoben. Ein „schnelles Testen“ wird veranlasst, indem der gesamte Test-Datensatz der 100 Domains „schnell“ (also ohne manuelle Kontrolle) über die Anwendung analysiert wird. Nach erfolgreicher Ausführung wird ein zweiter manueller Test-Durchlauf gestartet. Hierbei wird erneut wie beim ersten Test-Durchlauf der gesamte Test-Datensatz manuell analysiert und die generierten Speicherdateien der einzelnen Online-Shops mit den tatsächlich manuell ermittelten Einschätzungen verglichen. Die daraus resultierenden Ergebnisse werden erneut dokumentiert. Die hieraus entstandenen Gesamtergebnisse werden im Folgenden Kapitel aufgelistet und verdeutlicht.

Während der beiden manuellen Tests sind zudem einige neue Keywörter (sowohl positive/negative als auch GDPR-Keywörter) auf den verschiedenen Online-Shops erkennbar gewesen. Nach entsprechender Dokumentation und Filterung werden diese ebenfalls in die Anwendung integriert. Die hieraus resultierende verbesserte Erkennungspräzision sorgt kombiniert mit dem aktualisierten Programmcode für eine erhebliche Verbesserung der Anwendung.

5.1.2 Wie präzise arbeitet die entwickelte Anwendung?

Durch das mehrmalige Testen der Anwendung, befindet sich diese aktuell (basierend auf Durchlauf eins) in einem aktualisierten Zustand. Die ermittelten Ergebnisse werden im Folgenden zusammengefasst und näher erläutert.

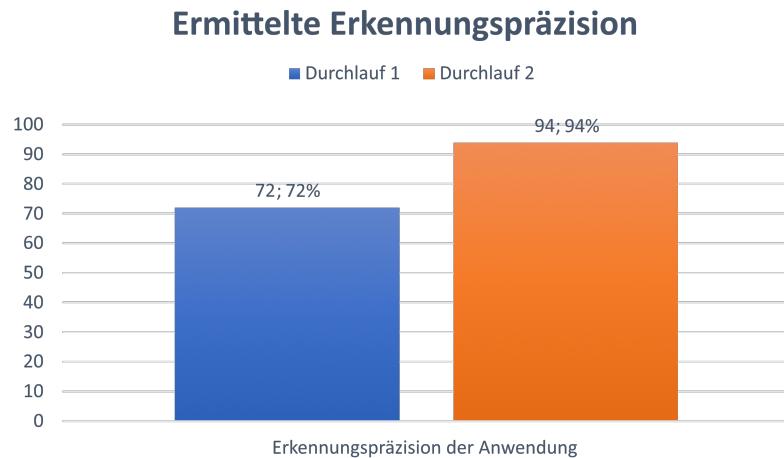


Abbildung 5.1: Ermittelte Erkennungspräzision vor/nach Integrierung der Ergebnisse

Abbildung 5.1 zeigt die allgemeine Erkennungspräzision vor sowie nach der Integrierung der in Durchlauf eins ermittelten und benötigten Änderungen. Die Definition der Erkennungspräzision beschreibt hierbei die korrekte Anwendungserfassung der verschiedenen Metriken, der verwendeten Cookie-Banner, der Cookie-Erfassung und Deklarierung sowie der erzeugten Resultate. Hierbei soll eine zu 100% sichere Erfassung dieser gelten.

Es zeigt sich, dass **72** (also auch **72%**) der analysierten 100 Online-Shops im ersten Durchlauf korrekt durch die Anwendung erfasst und deklariert worden sind. Bei der Analyse der restlichen 28 Online-Shops hat die Anwendung einzelne Metadaten oder sogar gesamte Resultate verfälscht dokumentiert.

Der auf den Test-Ergebnissen (integrierter aktualisierter Programmcode) basierende zweite Testdurchlauf zeigt eine Erkennungspräzision von **94%**, was somit einer Präzisionssteigerung von **22%** entspricht.

Zusätzliche automatische Test-Durchläufe beeinflussen dieses Ergebnis kaum bis gar nicht. Basierend auf diesem Ergebnis kann der vorliegende Programmcode in der Form angepasst werden, dass die daraus resultierende Erkennungspräzision 100% beträgt. Dieses Ergebnis kann dann allerdings nur mit Hilfe des verwendeten Test-Datensatzes belegt werden. Bei der Einführung eines größeren Datensatzes wird dieser „künstliche“ Wert wieder verringert und somit verfälscht.

Die aktualisierte Anwendung beschreibt zum jetzigen Zeitpunkt also ein Werkzeug, welches die Cookie-Informationen (bezüglich der Cookie-DSGVO-Konformität) von Shopify-Online-Shops mit einer **92-prozentigen Erkennungspräzisions-Wahrscheinlichkeit** erfasst, verarbeitet und analysiert. Die Erhöhung der Erkennungspräzision wird im Laufe dieser Arbeit als Diskussionsthema erneut aufgegriffen (siehe Kapitel 6.2)

Zwischenresultate

Die bei den verschiedenen Tests generierten Ergebnisse und Resultate werden schon durch erste Forschungsergebnisse ergänzt. Während der Analyse der 100 Shop-Domains sind bereits 100 erste Speicherdateien entstanden, welche nach Integrierung des aktualisierten Programmcodes sowie der neuen Keywörter zuverlässige Aussagen über die DSGVO-Konformität einzelner Online-Shops machen können. An diesem Punkt stellt sich allerdings die Frage, wie diese Speicherdateien ausgelesen und veranschaulicht werden sollen. Einige wenige Speicherdateien können hier problemlos manuell ausgelesen werden. Bei den jetzigen 100 Speicherdateien oder einer noch größeren Menge muss allerdings ein System entwickelt werden, welches diese entsprechend auf den Inhalt analysiert und die generierten Informationen zusammenfasst. Eine Lösung für dieses Problem wird noch im Laufe dieser Arbeit näher erläutert (siehe Kapitel 5.5).

5.2 Analyse der Berechnungsgeschwindigkeiten

Bei den Ausführungen der verschiedenen Tests sind neben den eigentlichen Daten zur Verbesserung der Anwendung zudem individuelle Berechnungsgeschwindigkeiten entstanden. Diese sind nach einzelnen Analysen erfasst und dokumentiert worden. Durch den Mittelwert der ermittelten Berechnungsgeschwindigkeiten wird im späteren Verlauf eine genauere Prognose bezüglich einer Ausführung basierend auf einer größeren Menge an Online-Shop-Domains aufgestellt. Die Folgende Tabelle listet die Berechnungsgeschwindigkeiten des ersten Tests auf.

Tabelle 5.1: Berechnungsgeschwindigkeiten in Sekunden - Test 1

22,0	22,0	20,5	20,4	21,7	20,8	20,6	21,8	22,3	20,4
22,4	20,0	22,4	21,3	20,2	20,1	20,1	20,3	20,8	21,4
21,3	22,1	21,5	21,5	22,1	20,7	21,7	21,0	21,0	22,0
21,4	21,0	21,3	22,4	22,4	20,4	21,4	20,9	21,0	21,5
20,0	21,3	20,9	21,8	21,8	20,8	22,4	21,0	20,6	21,1
20,3	21,0	21,4	20,6	20,3	20,1	20,4	20,9	22,1	22,2
22,5	21,7	22,0	20,8	20,1	21,6	21,9	21,8	21,4	22,1
21,1	21,5	22,4	21,8	21,0	21,9	20,2	21,7	21,7	22,4
20,8	21,9	21,5	22,3	21,6	20,2	20,3	21,6	22,5	20,2
20,5	20,8	20,8	20,3	21,8	20,6	21,2	20,8	21,2	21,6

Die in Tabelle 5.1 aufgelisteten Werte sowie der sich daraus bildende **Mittelwert** von **21,3 Sekunden** beschreiben eine erste Version der erfassten individuellen Berechnungsgeschwindigkeiten. Das Ziel hierbei ist die Optimierung dieser Werte anhand von verschiedenen Anpassungen der vorhandenen Ausführungsmethoden sowie der auszuführenden Anwendungsstruktur. Die Integrierung der verschiedenen Anpassungen wird in Kapitel 5.3 näher erläutert.

5.3 Anpassung der Ausführungsmethoden und der Anwendungsstruktur

Die Optimierung der Anwendung zur Verbesserung der ermittelten Berechnungsgeschwindigkeiten einzelner Online-Shop-Domains lässt sich durch eine einfache Anpassung der „Pausenzeiten“ zwischen den einzelnen Ausführungsmethoden einleiten. Zudem sorgen integrierte „try/catch“ Abfangmechanismen dafür, dass einzelne Prozesse bei fehlerhaftem Verhalten nicht direkt abgebrochen werden müssen, sondern dementsprechend reagiert werden kann. Bei diesen Mechanismen wird außerdem darauf geachtet, dass Speicherdateien, welche Fehlerhaft befüllt, bzw. welche aus verschiedenen Gründen überhaupt nicht mehr existieren dürfen, dementsprechend entfernt werden. Durch diesen zusätzlichen Prozess werden benötigte Ressourcen eingespart, was wiederum zu einer Verbesserung der Berechnungsgeschwindigkeiten führt.

Zusätzlich zu den vorher genannten Anpassungen der Ausführungsmethoden, ist zudem ein sogenannter Headless-Mode in die Anwendung integriert worden. Dieser Selenium-eigene Modus kann als „Option“ zum verwendeten Webdriver angehangen werden, was zum Verzicht der visuellen Ausgabe des Browsers gegenüber dem Anwender führt. Der Headless-Mode lässt sich also so zusammenfassen, dass die gesamte Anwendung „im Hintergrund“ ausgeführt wird, wodurch Ressourcen für die Visualisierung eingespart werden können, was wiederum zu einer Verbesserung der Berechnungsgeschwindigkeiten führt.

```
use_headless_mode = True
if use_headless_mode:
    options = Options()
    options.add_argument("--headless")
    options.add_argument("--window-size=1920,1200")
    options.add_argument('--disable-blink-features=AutomationControlled')
    driver = webdriver.Chrome(executable_path=r"chromedriver.exe", options=options)
else:
    options = Options()
    options.add_argument("--window-size=1920,1200")
    options.add_argument(' --disable-blink-features=AutomationControlled')
    driver = webdriver.Chrome(executable_path=r"chromedriver.exe", options=options)
```



Abbildung 5.2: Integrierung des Headless-Modes

Abbildung 5.2 zeigt die unkomplizierte Integrierung des Headless-Modes in die Anwendung. Hierbei wird zu den bisherigen „Options“ des Webdrivers ein Argument namens „- -headless“ angehangen.

Durch die Verwendung dieses Arguments wird auf die Nutzung einer visuellen Ausgabe verzichtet. In der Anwendung selbst existiert die Option, den Headless-Mode entsprechend ein- und auszuschalten. Hierdurch besteht weiterhin die Möglichkeit, die Anwendung mit einer entsprechenden visuellen Ausgabe zu verwenden.

Resultate

Durch die Anpassungen der Anwendungsmethoden sowie der Integrierung des Headless-Modes, befindet sich die Anwendung nun in einem aktualisierten Zustand, wodurch es sich anbietet, den in Kapitel 5.2 durchgeführten Test (Test 1) zu wiederholen (Test 2). An dieser Stelle wird noch einmal erwähnt, dass diese beiden Tests mit den ab Kapitel 5.1 erwähnten Tests gleichzustellen sind, da die Entwicklung parallel stattgefunden hat.

Tabelle 5.2: Berechnungsgeschwindigkeiten in Sekunden - Test 2

19,0	18,2	19,0	18,2	19,7	19,2	18,0	19,8	18,3	19,8
18,1	19,1	18,2	18,3	19,6	19,4	19,5	18,2	18,9	18,7
19,4	20,3	18,2	19,6	20,5	19,2	19,3	20,2	18,7	19,2
20,2	19,7	18,8	18,9	18,6	19,0	18,8	18,7	19,5	18,6
18,5	19,1	20,1	19,9	19,0	19,4	19,1	19,9	18,2	20,3
20,4	19,5	18,1	18,3	18,4	18,3	20,5	18,3	20,3	20,1
19,0	19,9	19,0	20,5	18,2	20,3	18,3	18,2	18,4	20,3
19,4	18,4	18,3	18,1	19,5	18,1	20,2	18,2	20,2	20,3
19,3	18,2	19,2	18,9	20,1	18,8	18,7	20,3	20,0	20,5
20,0	18,9	18,6	18,1	18,0	20,4	19,1	18,3	18,5	18,8

Tabelle 5.2 zeigt die aus Test 2 resultierenden Berechnungsgeschwindigkeiten. Hierbei kann eine deutliche Verbesserung festgestellt werden. Der resultierende **Mittelwert** von **19,1 Sekunden** verspricht eine deutliche Berechnungsgeschwindigkeits-Optimierung der späteren Analyse einer größeren Menge an Online-Shop-Domains.

5.4 Erzeugung und Nutzung eines größeren Datensets

Die bisher erzeugten Ergebnisse sollen dazu verwendet werden, ein allgemein größeres Datenset an Online-Shop-Domains zu analysieren. Anhand dieser erweiterten Analyse soll ein grobes Abbild der DSGVO-Konformität der deutschen Shopify-Online-Shops entstehen. Damit diese Analyse durchgeführt werden kann, muss zuerst eine dementsprechend größere Menge an Online-Shop-Domains generiert und erfasst werden.

Die Erzeugung der Domains ist hierbei identisch zu der verwendeten Methode aus Kapitel 4.7. Über die Plattform myip.ms wird anhand verschiedener Eingabeparameter wie der Top-Level-Domain „.de“ sowie den passenden Shopify IP-Adressen eine Liste an deutschen Shopify-Domains erzeugt, welche für die durchzuführende Analyse verwendet werden kann. Diese zurückgegebene Liste weist insgesamt **3.288 deutsche Shopify-Domains** auf, diese gilt es für die Analyse zu erfassen.

Die über myip.ms erzeugte Auflistung der Shop-Domains kann über eine entsprechende zur Verfügung gestellte Download-Funktion direkt von myip.ms heruntergeladen werden. Die hier erzeugte Excel-Datei ist identisch mit der im Browser ermittelten Auflistung der verschiedenen Shop-Domains (siehe Abbildung 4.5). Über verschiedene Excel-Formatierungen kann die vorliegende Excel-Datei so umstrukturiert werden, dass lediglich die Domain-Namen-Spalte verbleibt und die darin enthaltenen Domain-Namen mit dem Vorlauf „<https://www.>“ ergänzt werden. Die aktualisierte Datei wird anschließend als einfache Textdatei exportiert und in das Verzeichnis der entwickelten Anwendung geladen. Diese erhält hierdurch Zugriff auf eine Auflistung an aufrufbaren Domains. Über eine einfache Schleife können hierbei einzelne Domains erfasst und aufgerufen werden.

Die exportierte Textdatei weist zu diesem Zeitpunkt alle **3.288** Shop-Domains auf. Hierbei zeigt sich, dass das Extrahieren von Shop-Domains sowie die entsprechenden darauffolgenden Anweisungen des Hauptskriptes, zu einer „Überlastung“ des Hauptspeichers führen kann. Aus diesem Grund wird die vorhandene Groß-Text-Datei in jeweils 400 Shop-Domain-große Textdateien aufgeteilt, welche während dem Analyse-Prozess nacheinander durchlaufen werden (mehr zu diesem Problem in Kapitel 5.7).

Kalkulation: Wie viel Zeit nimmt die Hauptanalyse in Anspruch?

Anhand der vorherigen Kapitel (ab 5.2), welche sich mit den allgemeinen Berechnungsgeschwindigkeiten einzelner Analysen befasst haben, kann nun eine grobe Schätzung der durchzuführenden Haupt-Analyse ermittelt werden. Hierbei wird der optimierte Mittelwert der ermittelten Berechnungsgeschwindigkeiten aus Kapitel 5.3 verwendet, um eine entsprechende Kalkulation der benötigten Zeit zu garantieren.

Basierend auf **3.288 zu analysierenden Online-Shops** sowie des ermittelten Mittelwerts der Berechnungsgeschwindigkeit für einen einzelnen Online-Shop von **19,1 Sekunden**, ergibt sich eine geschätzte Dauer von ~ **62.800 Sekunden**, also **17,4 Stunden**.

Nach Beginn der Ausführung arbeitet die Anwendung vollkommen eigenständig, indem sie

- die einzelnen Textdateien nacheinander öffnet,
- die dort enthaltenen Shop-Domains extrahiert, erfasst und erneut formatiert,
- diese Shop-Domains über den Webdriver öffnet und wartet, bis diese vollständig geladen sind
- und anschließend die entsprechende DSGVO-Konformität analysiert.

Die ermittelten Ergebnisse sowie die Methoden zur Extraktion und Veranschaulichung relevanter Informationen, werden in den Folgenden Kapiteln näher erläutert.

5.5 Extraktion relevanter Informationen

Nachdem das Groß-Datenset analysiert ist, befinden sich **2.422 Speicherdateien** im Anwendungsverzeichnis. Anhang [Aufbau einer Speicherdatei] zeigt hierbei erneut den Aufbau sowie die Struktur einer Speicherdatei. Auffällig nach der Analyse sind **866 Shop-Domains**, welche durch keine Speicherdatei abgebildet werden. Der Grund dafür basiert auf der Annahme, dass diese Online-Shops nicht erreichbar (bspw. Wartungsarbeiten, nicht mehr online verfügbar oder sonstige Fehler) sind und diese deswegen auch nicht analysiert werden können. Die integrierte Screenshot-Funktionalität der Anwendung bestätigt diese Annahme. Abbildung 5.3 zeigt verschiedene nicht erreichbare und analysierbare Online-Shops.

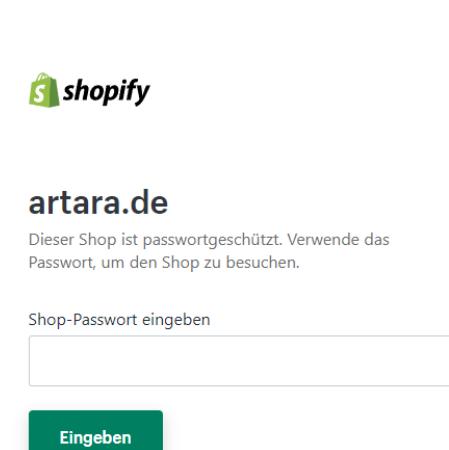
Die sich im Anwendungsverzeichnis befindenden **2.422 Speicherdateien** enthalten zu diesem Zeitpunkt Informationen über die DSGVO-Konformität über einzelne Shopify Online-Shops. Damit nun ein allgemeines Abbild der DSGVO-Konformität aller analysierten Online-Shops erstellt werden kann, wird ein weiteres Skript benötigt, welches diese vorhandenen Speicherdateien analysiert und die erfassten Ergebnisse zusammenfassend veranschaulicht. Dieses Skript, dessen Funktionsweise sowie die dazu passende Veranschaulichung über die Anwendung Microsoft Excel, werden im Folgenden näher erläutert.

5 Erzeugung und Evaluation



The screenshot shows a warning message from **united-domains** stating: "Ups! Irgendwas ist schief gelaufen." A blue box on the right says: "Diese neue Domain wurde im Kundenauftrag registriert." Below this, there's a section titled "Was ist passiert?" with the message "This store is unavailable". To the right, a sidebar provides information about why the page is shown and links to their services.

(a) Ein nicht verfügbarer Shopify-Shop



The screenshot shows a **shopify** logo and the URL **artara.de**. It says: "Dieser Shop ist passwortgeschützt. Verwende das Passwort, um den Shop zu besuchen." Below is a password input field and a green "Eingeben" button.

(c) Gewarteter Shopify-Shop

Trendanchor



The screenshot shows the **Trendanchor** logo and the text "Öffnet bald". It includes a "Shop-Passwort eingeben" input field, a "Deine E-Mail" input field, and a "Informiere mich" button. Below the input fields, it says "Finden Sie heraus, wann wir öffnen:" and "Wir sind bald wieder für euch da!". There are social sharing buttons for Facebook, Twitter, and Google+.

(d) Gewarteter Shopify-Shop

Abbildung 5.3: Nicht erreichbare und analysierbare Online-Shops

Das entwickelte Skript namens „ExtractData.py“ durchläuft nach der Ausführung die vorhandenen generierten Speicherdateien, um die darin enthaltenen Informationen am Ende der Ausführung zusammenfassend darzustellen. Hierfür durchläuft es die einzelnen Zeilen einer Speicherdatei, extrahiert dabei die vorhandenen Werte und speichert diese in entsprechenden Variablen. Diese werden anschließend zusammenfassend über eine entsprechende Konsolenausgabe ausgegeben.

Für eine bessere Veranschaulichung werden in der Konsole zudem noch entsprechende Aussagen über Domains gemacht, welche Bedingungen nicht erfüllen. Häufig geschieht dies über die Angabe der Werte direkt oder über einen berechneten Prozentwert. Zur Veranschaulichung zeigt Anhang [Konsolenausgabe „ExtractData“-Skript (deutsche Ausgabe)] hierbei die erzeugte Ausgabe des ExtractData Skripts.

Für eine bessere Darstellung der ermittelten Hauptergebnisse werden die ermittelten Daten der Hauptanalyse über eine externe Anwendung wie Microsoft Excel grafisch aufbereitet. Die hier erzeugten Diagramme sowie die Ergebnisse allgemein werden im Folgenden Kapitel näher erläutert.

5.6 Resultierende Ergebnisse

In diesem Kapitel werden die erzeugten Ergebnisse der Hauptanalyse dargestellt und erläutert. Diese Ergebnisse basieren auf insgesamt **3.288 Online-Shops**, wovon **2.422** analysiert worden sind.

Grundlegende Ergebnisse

Neben den eigentlichen Hauptergebnissen sind verschiedene Merkmale bezüglich der DSGVO-Konformität überwacht worden, diese werden im Folgenden kurz zusammengefasst. Merkmale beschreiben nur eine „Annahme“ bzw. eine Schätzung, ob etwas in einem Online-Shop existent ist.

Die ermittelten Ergebnisse der Analyse zeigen, dass fast jeder Online-Shop (**99,05%**) unterschiedliche Cookie-Banner Merkmale im Quellcode aufweist, zudem besitzen diese ebenfalls Merkmale für einen „Akzeptieren“-Knopf. Hieraus darf allerdings nicht die Schlussfolgerung entstehen, dass diese Online-Shops einen Cookie-Banner integriert haben, die Folgenden Unterkapitel schaffen hier Klarheit.

Neben den Merkmalen eines „Akzeptieren“-Knopfes, fallen die Merkmale eines „Ablehnen“-Knopfes deutlich geringer aus. Während **1,2%** der Online-Shops, Merkmale für einen einfachen „Ablehnen“-Knopf aufweisen, existieren bei **21,8%** der Online-Shops Merkmale für einen GDPR-konformen Auswahlmechanismus. Diese ersten Ergebnisse zeigen, dass viele Online-Shops zwar Merkmale für einen „Akzeptieren“-Knopf aufweisen, in **77%** der Fälle allerdings keine Merkmale für einen „Ablehnen“-Knopf vorhanden sind.

Verwendung eines GDPR-konformen Cookie Banners

Abbildung 5.4 zeigt das Nutzungsverhalten der Online-Shops eines DSGVO-konformen Cookie-Banners. Hierbei zeigt sich, dass **65,15%** der analysierten Online-Shops einen DSGVO-konformen Cookie-Banner verwenden. Allerdings muss hier zudem noch beachtet werden, dass diese Information nur die „Oberfläche“ des Cookie-Banners betrachtet, die eigentliche DSGVO-Konformität lässt sich zudem anhand der geladenen Cookies bestimmen.

NUTZUNG VON GDPR-KONFORMEN COOKIE-BANNERN

■ Nutzung eines GDPR-konformen Cookie-Banners ■ Nutzung sonstiger Cookie-Banner

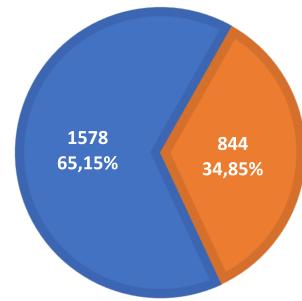


Abbildung 5.4: Anzahl der Online-Shops, welche DSGVO-konforme Cookie-Banner verwenden

Anbieten einer Ablehnungsmöglichkeit

Bei den tatsächlich verwendeten Ablehnungsmöglichkeiten zeigt sich laut Abbildung 5.5, dass **78,49%** der analysierten Online-Shops dem Nutzer keine DSGVO-konforme Ablehnungsmöglichkeit anbieten. Die vorher aufgestellten Merkmale für einen „Ablehnen“-Knopf haben also korrekterweise eine erste Andeutung über das „Ablehnungsverhalten“ der analysierten Online-Shops gemacht.

ABLEHNUNGSMÖGLICHKEIT

■ Bieten keine „Ablehn“-Option ■ Bieten „Ablehn“-Option

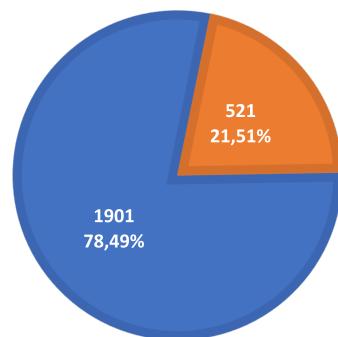


Abbildung 5.5: Anzahl der Online-Shops, welche dem Nutzer eine Möglichkeit zum Ablehnen eines Cookie-Banners anbieten

Sondermerkmal

Mit dem Sondermerkmal ist die Situation gemeint, in welcher ein Online-Shop verschiedene Cookie-Banner-Merkmale im Quellcode besitzt, diese allerdings mit keinem sichtbaren Cookie-Banner übereinstimmen. Abbildung 5.6 zeigt, dass **39,35%** der analysierten Online-Shops verschiedene Cookie-Banner-Merkmale besitzen, dem Nutzer allerdings keinen entsprechenden Cookie-Banner anzeigen.



Abbildung 5.6: Anzahl der Online-Shops, welche Cookie-Merkmale im Quellcode besitzen aber dem Nutzer keinen Cookie-Banner präsentieren

Durch die in die Anwendung integrierte Screenshot-Funktion kann so ein beschriebener Online-Shop in Abbildung 5.7 betrachtet werden.

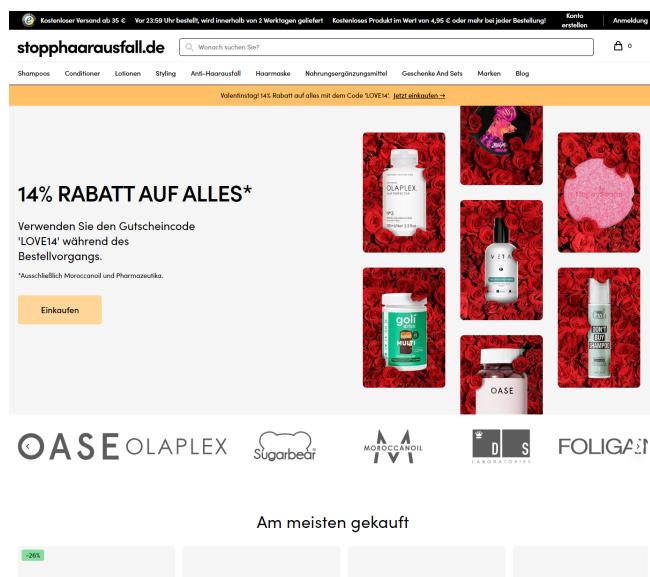


Abbildung 5.7: Online-Shop mit vorhandenen Cookie-Merkmalen und fehlendem Cookie-Banner

Nutzung von Third-Party-Cookies

Bei der ungenehmigten Nutzung von Cookies laden laut Abbildung 5.8 **58,51%** der analysierten Online-Shops zu Beginn unerlaubt Third-Party-Cookies in das Browser-Cookie-Verzeichnis des Nutzers. Hierbei muss erneut erwähnt werden, dass der Nutzer zu diesem Zeitpunkt keine entsprechende Einwilligung für diese Integrierung erteilt hat, häufig werden diese Cookies sogar noch vor dem eigentlichen Cookie-Banner geladen.

THIRD-PARTY COOKIE-NUTZUNG

■ Nutzen zu Beginn unerlaubt Third-Party Cookies
 ■ Nutzen zu Beginn nur essentielle Cookies

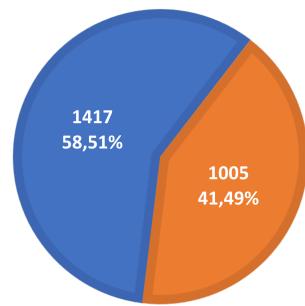


Abbildung 5.8: Anzahl der Online-Shops, welche zu Beginn ungenehmigt Third-Party-Cookies laden

Ungenehmigte Third-Party Cookie-Nutzung

Durch Abbildung 5.9 lassen sich die verschiedenen Third-Party-Cookies einsehen, welche ungenehmigt in das Browser-Cookie-Verzeichnis des Nutzers geladen werden. Es zeigt sich, dass fast die Hälfte der analysierten Online-Shops (**49,3%**) einen Google Service als Third-Party-Cookie ungenehmigt nutzen (zur Nachverfolgung des Nutzers oder aus sonstigen Monetären Zwecken). Für die weitere Nachverfolgung des Nutzers laden **37,7%** der analysierten Online-Shops ungenehmigt den Facebook Pixel und **12,14%** den Pinterest Tag. Als Beispiel für eine Third-Party-Anwendung laden **19,69%** der analysierten Online-Shops Klaviyo, um eine einfache Kommunikation mit dem Nutzer nach dem Shop-Besuch zu ermöglichen (E-Mail oder SMS).

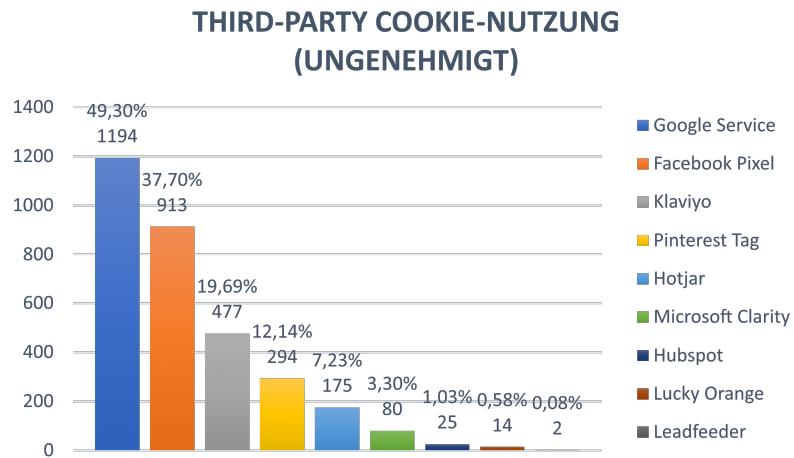


Abbildung 5.9: Ungenehmigte geladene Third-Party-Cookies

Respektierung der Nutzerentscheidungen

Abbildung 5.10 basiert auf dem zusätzlichen Szenario, in welchem der Nutzer die Möglichkeit hat, über die Integrierung der verschiedenen Cookies zu entscheiden. **59,83%** der analysierten Online-Shops respektieren dabei nicht die Entscheidung des Nutzers. Fast 60% „übergehen“ also den Nutzer, indem sie geplante Cookies unabhängig von der Entscheidung des Nutzers in dessen Browser-Cookie-Verzeichnis laden.



Abbildung 5.10: Anzahl der Online-Shops, welche die Nutzerentscheidungen respektieren

Ermittelte Cookie-Mengen

Anhand von Abbildung 5.11 lassen sich die verschiedenen Mengen der ermittelten Cookies einsehen. Hierbei ist eine Ausweitung in den höheren Bereich zu erkennen, welcher nach der Akzeptierung des Cookie-Banners zwischen **10** und **30** Cookies liegt. Auffällig hierbei sind **47 Online-Shops**, welche zwischen **40** und **50** Cookies nach der Akzeptierung des Cookie-Banners zusätzlich verwenden.

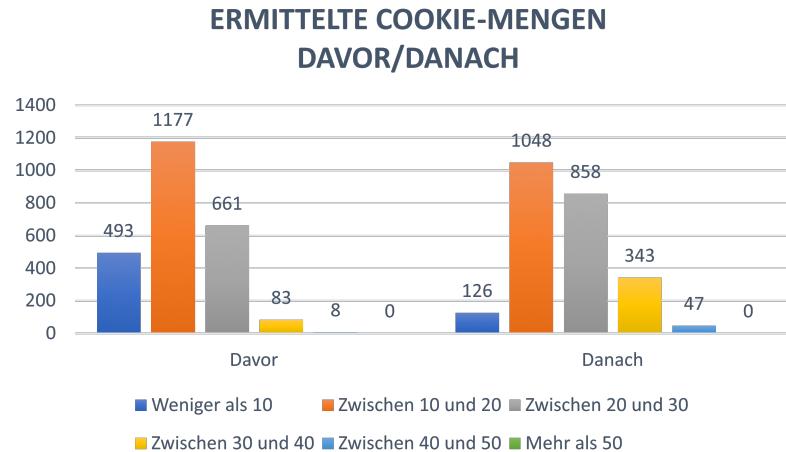


Abbildung 5.11: Anzahl der Online-Shops, welche die jeweiligen Cookie-Mengen vor und nach der Akzeptierung des Cookie-Banners besitzen

Menge der hinzugefügten Cookies

Abbildung 5.12 ist eine Zusammenfassung der Menge der geladenen Cookies (nach der Akzeptierung des Cookie-Banners). **83,07%** der analysierten Online-Shops laden weniger als **10 Cookies**, **12,8%** hingegen zwischen **10** und **20 Cookies** und **3,96%** zwischen **20** und **30 Cookies**.

Dieser hohe Wert (**83,07%**) stammt unter anderem aus dem Ursprung, dass viele dieser Online-Shops bereits zu Beginn die meisten Cookies (häufig ungenehmigt) geladen haben, weswegen nach der Akzeptierung nur noch wenige benötigt werden.

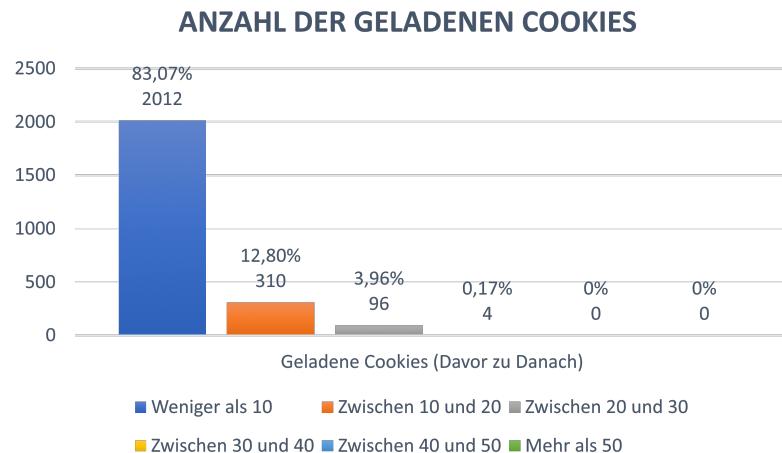


Abbildung 5.12: Anzahl der Online-Shops, welche die jeweiligen Cookie-Mengen nach der Akzeptierung des Cookie-Banners hinzugefügt haben

DSGVO-Konformität

Die sich hieraus bildende allgemeine DSGVO-Konformität aller analysierten Online-Shops lässt sich in Abbildung 5.13 ablesen. Fast 70% (**69,32%**) der analysierten **2.422 Online-Shops** sind nicht DSGVO-konform. Der Grund hierbei liegt neben fehlenden optischen DSGVO-konformen Cookie-Bannern häufig auch bei der ungenehmigten Verwendung von Cookies. Nutzer werden häufig zu Beginn aber auch nach einer entsprechenden Entscheidung im späteren Verlauf „übergangen“, indem Cookies unabhängig von der Entscheidung des Nutzers geladen werden.

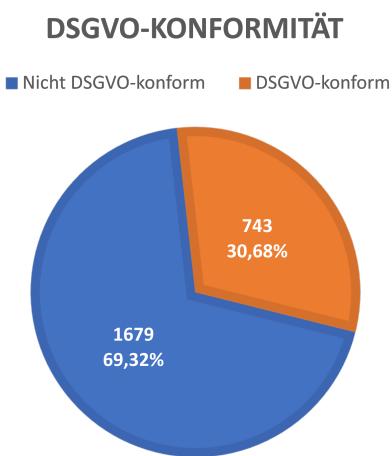


Abbildung 5.13: Anzahl der Online-Shops, welche DSGVO-konform sind

5.7 Probleme

Bei der Entwicklung der Anwendung sind verschiedene Probleme aufgetreten, welche nicht (oder nur teilweise/nicht optimal) gelöst worden sind. Diese werden im Folgenden kurz dargestellt und erläutert.

Nicht aufrufbare Online-Shops

Beim Aufrufen eines Online-Shops kann jederzeit das Szenario eintreten, dass dieser in diesem Moment aus unterschiedlichsten Gründen (Wartungsarbeiten/sonstige Ursachen und Fehler) nicht erreichbar ist. Die Anwendung muss in der Lage sein, dieses Szenario nach dem Laden des Online-Shops automatisch zu erkennen und dementsprechend darauf zu reagieren.

```
# Check if website is still available
if website_warning in html_source or \
    website_warning_2 in html_source or \
    website_warning_3 in html_source or \
    website_warning_4 in html_source or \
    website_warning_5 in html_source or \
    website_warning_6 in html_source:
    print("Website not available!")
    print("Skip website...")
    offline_websites_amount += 1
    continue
# Warnings
website_warning = "Only one step left!"
website_warning_2 = "is currently unavailable."
website_warning_3 = "Dies ist keine sichere Verbindung"
website_warning_4 = "Domain erwerben"
website_warning_5 = "Be the first to know when we launch"
website_warning_6 = "Die Website ist nicht erreichbar"
```

(a) Filterung von nicht erreichbaren Online-Shops (b) Identifizierbare Sätze in nicht aufrufbaren Online-Shops

Abbildung 5.14: Identifizierung von nicht aufrufbaren Online-Shops

Abbildung 5.14 zeigt die entsprechende Implementierung für dieses Problem in die bestehende Anwendung. Hierbei wird eine Menge an typischen Fehlermeldungen für einen nicht erreichbaren Online-Shop definiert, welche nacheinander mit dem vorhandenen Quellcode abgeglichen werden. Bei einer positiven Übereinstimmung wird der derzeitig zu analysierende Online-Shop übersprungen, was dazu führt, dass die nächste Domain eines Online-Shops geladen und daraufhin analysiert wird.

Automatische Identifizierung von neuen Keywörtern

Die Anwendung arbeitet mit einer vordefinierten Menge an Keywörtern, welche manuell (z.B während der Ausführung der verschiedenen Tests) aktualisiert werden müssen. Hierbei bietet es sich an, diesen Prozess zu automatisieren, indem die Anwendung neue Keywörter in neuen Online-Shops automatisch erkennt und aufnimmt.

Der Grund für den Verzicht auf diese Funktionalität liegt in der Tatsache, dass die Anwendung ein bestimmtes vordefiniertes Datenset an Online-Shop-Domains für die verschiedenen Berechnungen besitzt. Anhand von diesem Datenset lassen sich bereits alle gängigen Keywörter manuell ermitteln und in die Anwendung integrieren.

Eine Funktionalität zur automatischen Erfassung von Keywörtern spielt hier eine untergeordnete Rolle und ist in diesem Szenario nicht notwendig (anders als bspw. in einem Szenario mit einer 5(+)-steligen Menge an Online-Shop-Domains).

Fehlerhafte Erkennung von Cookie-Banner-Knöpfen

Ein weiteres Problem, welches bei der Entwicklung der Anwendung aufgetreten ist, sind die verschiedenen Szenarien, in welchen gefundene Keywörter in Online-Shops nicht zu Cookie-Banner-Knöpfen führen, sondern zu sonstigen Elementen im analysierten Online-Shop. Ein Beispiel-Szenario beschreibt bspw. die Situation, in welcher das Keywort „ok“ in einem einfachen Text-Element gefunden worden ist, welches das englische Wort „look“ enthält. Die Anwendung versucht in diesem Fall auf dieses Text-Element zu klicken, was letztendlich dazu führt, dass der Cookie-Banner unberührt bleibt und die darauffolgenden generierten Resultate verfälscht werden.

Für eine präzisere Erkennung wird wie bereits im Laufe dieser Arbeit erwähnt die „re“-Bibliothek verwendet. Zudem werden beide Parteien (Keywort und analysiertes Web-Element sowie dessen Inhalt) „gleichgestellt“, indem sie über die String-Methode „.casefold()“ (ähnlich wie die String-Methode „.lower()“) zu Kleinbuchstaben „verkleinert“ werden. Hierdurch soll ein gleichwertiger Vergleich garantiert werden. Um sicherzustellen, dass die Anwendung lediglich vorhandene Cookie-Banner-Knöpfe betätigt, wird zudem überprüft, ob das ermittelte Element auch ein klickbarer Knopf ist. Hierfür werden die Knopf-Methoden „.is_enabled()“ sowie „.is_displayed()“ für die Überprüfung verwendet.

Hauptspeicher-Probleme (eine große Domain-Datei)

Bei der Speicherung der verschiedenen ermittelten Online-Shop-Domains, sowie bei den ersten Ausführungen der Anwendung hat sich herausgestellt, dass eine „Großdatei“ mit Domains zu einer Überlastung des Hauptspeichers führt. Die Anwendung ist in diesem Szenario nicht mehr in der Lage, die vorgeschriebene Analyse für die **3.288 Online-Shop-Domains** durchzuführen.

Für einen reibungslosen Ablauf der Anwendung wird auf eine generelle „Großdatei“ verzichtet. Stattdessen werden **acht einzelne Dateien** verwendet, welche maximal **400 Domains** enthalten. Zudem wird die Anwendung nach jedem Durchlauf einer einzelnen Domain-Datei manuell neugestartet, indem die nächste Domain-Datei manuell als Quellen-Verzeichnis für zu analysierende Domains deklariert wird. Hierdurch soll eine Leerung des vorhandenen Caches sichergestellt werden.

6 Diskussion

6.1 Anwendung

Grundlegend muss zuerst erwähnt werden, dass sich im Laufe dieser Arbeit das aktuelle Gesetz bezüglich des Datenschutzes (besonders in Deutschland) durch das TTDSG aber auch durch aktualisierte Bedingungen der DSGVO, erneut geändert hat. Die in dieser Arbeit aufgegriffenen Standards stimmen also nicht mehr gänzlich mit den aktualisierten Standards des aktuellen Datenschutzes überein.

Die ermittelten Ergebnisse der entwickelten Anwendung jedoch beschreiben nach wie vor die mangelnde Auseinandersetzung der Online-Shops mit der aktuellen Lage des Datenschutzes. Auch wenn die ermittelten Ergebnisse zu diesem Zeitpunkt nicht mehr die aktuelle Gesetzeslage widerspiegeln, zeigen sie dennoch den aktuellen Bezug der Online-Shops zu diesem speziellen Thema.

Die aktualisierten Bedingungen zur DSGVO-Konformität weisen auch darauf hin, dass zukünftige Forschungen diese als grundlegend betrachten sollten. Generell versprechen allgemeine Forschungen in Form von Analysen der DSGVO-Konformität über einen längeren Zeitraum, wertvolle Informationen über die aktuelle Lage des Datenschutzes. Die Beachtung der aktualisierten Bedingungen bestärken hierbei die Generierung dieser Informationen.

Die in dieser Arbeit entwickelte Anwendung ist sehr „empfindlich“ konstruiert worden. Generell bedeutet das, dass die Anwendung bei einem auffälligen Fund von Cookie-Cookie-Banner-Fragmenten im vorliegenden Quellcode, bevorzugt positiv über die Existenz von Cookies/Cookie-Banner urteilt. Der Grund hierfür liegt in der Tatsache, dass es besser ist, wenn die Anwendung etwas „zu früh“ erkennt, als wenn bspw. ein Cookie-Banner vorhanden ist, dieser aber nicht von der Anwendung erfasst und analysiert wird.

An diesem Punkt muss allerdings erwähnt werden, dass dieser ganze Mechanismus nicht „unterschätzt“ werden darf. Auch wenn die Anwendung immer zugunsten der Cookie-Banner urteilt, finden ausreichend Zwischenüberprüfungen für eine optimale Erkennungspräzision statt. Dennoch müssen einige wenige Punkte erwähnt werden, welche eventuell keine optimale Wahl für diese Anwendung gewesen sind bzw. welche für zukünftige Forschungen verbesserungswürdig sind.

Das entwickelte Keywort-System, mit welchem über vordefinierte Keywörter, Merkmale zu Cookies und Cookie-Banner identifiziert werden können, beschreibt grundlegend einen sinnvollen Ansatz für dieses Problem.

6 Diskussion

Zukünftige Forschungen allerdings sollten diesen Mechanismus durch einen gewissen Automatismus erweitern. Ein zusätzliches System, bzw. eine einfache künstliche Intelligenz kann hierbei unterstützend wirken, indem sie neue Keywörter bei einer Analyse entsprechend bewertet und in das vorhandene Keywort-Datenset aufnimmt. Bei einer Skalierung der verwendeten Domains entfällt hierdurch eine manuelle Erfassung von Keywörtern völlig.

Durch eine Skalierung muss zudem darauf geachtet werden, ein entsprechend stabiles Speicher-Datei-System zu entwickeln. Eine Speicherung von einzelnen Dateien pro entsprechender Domain ist in diesem Fall nicht empfehlenswert, da zu viele Speicherdateien zu unübersichtlich werden und schnell zu einer Verlangsamung der Anwendung führen. Für eine geringere Menge an Domains (wie in dieser Arbeit geschehen) ist dieser Mechanismus allerdings absolut ausreichend.

Grundlagen und Tools

Die allgemeine Erfassung von deutschen Shopify-Online-Shops hat im Laufe dieser Arbeit dank der IP-basierten Suche sehr gut funktioniert. Jedoch wird durch diese Methode immer nur ein Tages-aktueller Stand der Domains generiert. Nach einer bestimmten Zeit ist das generierte Datenset nicht mehr aktuell, was zu inaktiven oder nicht erreichbaren Online-Shops führt. Für zukünftige Forschungen wird diese Methode dennoch empfohlen, allerdings empfiehlt es sich, das verwendete Datenset anhand dieser Methode in regelmäßigen Abständen zu aktualisieren.

Wie bereits erwartet, hat Python zu einer deutlichen erwarteten Reduzierung des implementierten Codes geführt. Zudem war die optimale Anbindung verschiedener Bibliotheken (wie z.B die erwähnte „re“-Bibliothek als direkter Zugriff für die regulären Ausdrücke) sehr vorteilhaft.

Auch wenn das Selenium-Framework nicht direkt für diese Art von Anwendungen vorgesehen ist, war es im Laufe dieser Arbeit sehr gut auf die verschiedenen Probleme der zu entwickelnden Anwendung anwendbar. Die bereits durch Selenium in die Anwendung integrierten Mechanismen haben in einigen Situationen zu einer deutlichen Reduktion des benötigten Codes geführt. Allgemein hat sich dieses Framework für dieses Problem als außerordentlich nützlich erwiesen, was grundlegend auch zu schnellen Berechnungsgeschwindigkeiten geführt hat. Die in dieser Arbeit ermittelten Berechnungsgeschwindigkeiten für die Hauptanalyse haben zudem mit den tatsächlichen Berechnungsgeschwindigkeiten des Frameworks übereingestimmt. Hierbei sind keine größeren Unterschiede erkennbar gewesen.

6.2 Präzision und Genauigkeit

Zeiten

Die durchgeführte Hauptanalyse hat ca. 17 Stunden in Anspruch genommen. Hierbei entsteht die Frage, ob diese Zeit durch die Anpassung verschiedener Parameter in der Anwendung beeinflussbar ist. Der Hauptparameter, welcher dafür in Frage kommen kann, ist die allgemeine Wartezeit zwischen einzelnen Methoden, sowie direkt nach dem erfolgreichen Laden des Online-Shops. Im Laufe der Arbeit wurde dieser Wert immer wieder anhand verschiedener Tests reduziert, um die allgemeine Berechnungsgeschwindigkeit zu erhöhen. Hierbei zeigte sich allerdings, dass diese Reduktion zu einer häufigeren Fehlerhaften Erfassung von Informationen führte. Zudem sorgte die erhöhte Wartezeit nach dem Laden des Online-Shops für eine „sichere“ Erfassung. Hier konnte also verhindert werden, dass die durchzuführende Analyse startet, obwohl der zu analysierende Online-Shop noch nicht vollständig geladen war. Die angegebenen Zeiten beschreiben also optimale Werte für diese Anwendung.

Tests und verwendete Cookies

Die durchgeführten Test-Analysen haben sehr viel zu einer Optimierung der Erkennungspräzision beigetragen. Damit die allgemeine Erkennungspräzision bei einer Skalierung der verwendeten Online-Shop-Domains deutlich erhöht werden kann, bietet es sich an, eine größere Anzahl an zu erfassenden Cookies zu verwenden. Die in dieser Arbeit zum Teil erfassten Third-Party-Cookies (siehe 2.2.2) haben zwar für eine allgemeine Beurteilung der DSGVO-Konformität ausgereicht, bei einer größeren Menge von analysierten Domains wäre diese Cookie-Menge allerdings nicht mehr ausreichend, was zu einer Reduktion der Erkennungspräzision führen würde. Zudem bietet es sich an, stichprobenartige manuelle Überprüfungen einzelner Domain-Bereiche durchzuführen, um eine konstante Erkennungspräzision zu garantieren.

Nicht erreichbare Online-Shops

Im Laufe dieser Arbeit sind 866 Online-Shop-Domains auffällig gewesen, welche zu keinem erkennbaren Ergebnis geführt haben. Bei den meisten dieser Domains kann gesagt werden, dass diese aus unterschiedlichsten Gründen nicht erreichbar waren. Dennoch trifft dies nicht auf jeden dieser Online-Shops zu. Bei einigen war die durchgeführte Analyse zu ungenau, was zu einer fehlerhaften Erkennung der Cookies bzw. der Cookie-Banner geführt hat. Diese Gruppe von Online-Shops ist somit als „nicht erreichbar“ eingestuft worden, was wiederum fehlerhaft ist.

Zusammenfassend kann aber gesagt werden, dass diese Gruppe von Online-Shop-Domains einen kleinen Prozentsatz der Gesamt-Domain-Menge beschreibt. Dieser verfälscht die ermittelten Ergebnisse nur minimal.

6.3 Ergebnisse

Basierend auf den Ergebnissen der in dieser Arbeit genannten externen Quellen sowie den ermittelten Ergebnissen der in dieser Arbeit entwickelten Anwendung kann gesagt werden, dass diese sich sowohl unterschieden aber auch eine entscheidende Gemeinsamkeit besitzen. Obwohl die Ergebnisse nicht direkt miteinander übereinstimmen bzw. gewisse Ähnlichkeiten besitzen, beschreiben sie hervorragend das vorliegende Problem des aktuellen Datenschutzes im Internet. Zusammen mit den in dieser Arbeit aufgelisteten Ergebnisse bzw. Informationen beschreiben sie die „Nichtbeachtung“ sowie die Ignoranz der Webseiten-/Online-Shop-Betreiber zu diesem Thema. Auch wenn durch die Missachtung der Regulierungen verschiedene Mahnverfahren inklusive hoher Geldbußen drohen, ist das Verlangen nach einer zuverlässigen „Verfolgung“ des Nutzers über verschiedene Third-Party-Cookies und der damit zusammenhängende erhöhte Umsatz größer.

Bereits bei der Nutzung von GDPR-konformen Cookie-Bannern lässt sich erkennen, dass viele Online-Shop-Betreiber es dem Nutzer nicht ermöglichen wollen, entsprechende Cookie-Entscheidungen zu treffen. Einige dieser Online-Shops besitzen zwar entsprechende Cookie-Fragmente im Quellcode, zeigen dem Nutzer allerdings keinen entsprechenden Cookie-Banner an. Der Grund hierfür kann evtl. an der kurzzeitigen Deaktivierung/Aktivierung von diesen Cookie-Bannern liegen.

Generell respektieren mehr als die Hälfte der analysierten Online-Shops nicht die Entscheidung des Nutzers. Zudem laden gleichwertig viele unerlaubt Third-Party-Cookies in den Browserspeicher des Nutzers. Dieses „übergehen“ des Nutzers verdeutlicht erneut die Notwendigkeit der besagten Third-Party-Cookies. Anhand von Abbildung 5.9 kann hierbei der Trend der Google Services sowie des Facebook Pixels eingesehen werden, mit welchen eine optimale Verfolgung des Nutzers möglich ist.

Die resultierende DSGVO-Konformität bei deutschen Shopify Online-Shops fällt mit **30,68%** erschreckend gering aus. Hier zeigt sich erneut die Ignoranz der Betreiber zu diesem Thema. Zudem kann hier auch verdeutlicht und eingesehen werden, warum sich immer mehr Kanzleien auf diesen lukrativen Bereich spezialisieren. (Wischmeyer, 2018)

7 Fazit

Der Sinn dieser Arbeit bestand darin, eine Anwendung basierend auf einem Crawler zu entwerfen und zu entwickeln. Dieser Crawler sollte mit Hilfe des Frameworks Selenium sowie mit einer Leistungsstarken Programmiersprache (in dieser Arbeit Python) entwickelt werden. Die Aufgabe dieses Crawlers bestand darin, ein vordefiniertes Datenset mit deutschen Shopify Online-Shop-Domains auf deren DSGVO-Konformität zu analysieren. Die hier entstandenen Resultate sollten einen aktuellen Stand der DSGVO-Konformität der heutigen deutschen Shopify Online-Shops präsentieren. Basierend auf einem zusätzlich entwickelten Skript sollten diese generierten Daten kategorisiert, zusammengefasst und ausgegeben werden.

Grundlegend ist die Entwicklung eines Selenium basierten Crawlers gelungen. Dennoch ist diese entwickelte Version nicht final, sondern in wenigen Punkten verbessungswürdig. Es kann zudem gesagt werden, dass Selenium den zu implementierenden Code reduziert, viele fertige Methoden grundlegend zur Verfügung stellt und für eine erste stabil laufende Version mit ersten generierten Resultaten vollkommen ausreichend ist. Dennoch verbleibt das Problem, dass viele Aspekte der entwickelten Anwendung auf einer manuellen Überprüfung der Webseiten beruhen, welche analysiert werden sollen. Durch verschiedene Tests kann ein „Allgemeinbild“ der analysierten Webseiten erstellt werden. Die hier generierten Resultate sind zwar aussagekräftig, entsprechen allerdings nicht vollständig der Wahrheit. Selenium verbleibt als ein einfaches Testing-Framework, welches von einem Entwickler verwendet wird, um eine bestimmte Webseite zu analysieren. Im Normalfall existieren in diesem Szenario keine unbekannten. In dem Szenario dieser Arbeit müssen bspw. die verschiedenen Keywörter manuell erfasst und in die Anwendung integriert werden. Eine entsprechende Weiterentwicklung zu einem automatisierten System, welches die Keywörter automatisch erfasst, würde sich vom grundlegenden Selenium Framework immer weiter entfernen. Ein anderes Framework würde sich für dieses Problem wahrscheinlich als nützlicher erweisen.

In dem Fall, dass die entwickelte Anwendung bzw. das grundlegende Datenset an Online-Shop-Domains skaliert werden soll, sollte in jedem Fall auf die allgemeine Erkennungspräzision geachtet werden. Durch stichprobenartige Tests kann diese konstant gehalten werden, was zu einer gleichbleibenden Qualität der generierten Ergebnisse führt. Die Skalierung führt im Wesentlichen zwar zu spezifischeren Ergebnissen, eine direkte Tendenz der Online-Shop-Betreiber bezüglich der DSGVO-Konformität konnte allerdings bereits bewiesen werden. Zudem konnte gezeigt werden, dass mithilfe von Selenium ein entsprechender Crawler für dieses Problem implementiert werden kann.

7.1 Ausblick

Basierend auf dieser Arbeit sowie der entwickelten Anwendung konnte der aktuelle Stand der DSGVO-Konformität von deutschen Shopify Online-Shops dargestellt werden. Für das Szenario, in welchem dieser Crawler erneut für verschiedene Forschungen in diesem Bereich aufgegriffen wird, empfiehlt es sich, das enthaltene Keywort-System erneut zu überarbeiten. Dies kann in der Selenium Anwendung selbst geschehen oder extern über ein neues Framework. Letzteres wird hier allerdings empfohlen, da Selenium als ein eigenständiges Testing-Framework verbleiben soll.

Literaturverzeichnis

- [Ahmed 2022] AHMED, Zee: *Shopify vs WordPress [Comparison in 7 Key Areas]*. <https://onesmartsheep.com/web-design/shopify-vs-wordpress/>. Version: 2022. – Letzter Zugriff: 7. März 2022
- [Bollinger Dino 2022] BOLLINGER DINO, Cotrini Carlos Basin D. Kubicek Karel K. Kubicek Karel: *Automating Cookie Consent and GDPR Violation Detection*. <https://www.research-collection.ethz.ch/handle/20.500.11850/525815>. Version: 2022. – Letzter Zugriff: 04. Mai 2022
- [BW 2021] BW, Verbraucherportal: *Cookies - hilfreich oder gefährlich*. https://www.verbraucherportal-bw.de/Lde/Startseite/Verbraucherschutz/Cookies+-+hilfreich+oder+gefaehrlich_. Version: 2021. – Letzter Zugriff: 05. Mai 2022
- [Célestin Matte 2019] CÉLESTIN MATTE, Cristiana S. Natalia Bielova B. Natalia Bielova: *Do Cookie Banners Respect my Choice? Measuring Legal Compliance of Banners from IAB Europe's Transparency and Consent Framework*. <https://arxiv.org/abs/1911.09964>. Version: 2019. – Letzter Zugriff: 04. Mai 2022
- [Duncan 2022] DUNCAN, Katherine: *How Shopify Became the Go-To E-commerce Platform for Startups*. <https://www.entrepreneur.com/article/222967>. Version: 2022. – Letzter Zugriff: 7. März 2022
- [github.com/Perdu 2021] GITHUB.COM/PERDU: *Cookinspect*. <https://github.com/Perdu/Cookinspect>. Version: 2021. – Letzter Zugriff: 21. April 2022
- [Haucke 2022a] HAUCKE, Annika: *Das müssen Sie 2022 über die Datenschutzgrundverordnung wissen*. <https://www.e-recht24.de/datenschutzgrundverordnung.html>. Version: 2022. – Letzter Zugriff: 05. Mai 2022
- [Haucke 2022b] HAUCKE, Annika: *Das neue TTDSG: Neues bei Cookies und Co. für Webseitenbetreiber und Agenturen*. <https://www.e-recht24.de/artikel/datenschutz/12834-ttsg.html>. Version: 2022. – Letzter Zugriff: 05. Mai 2022
- [IntersoftConsulting 2018] INTERSOFTCONSULTING: *Datenschutz-Grundverordnung DSGVO*. <https://dsgvo-gesetz.de/>. Version: Mai 2018. – Letzter Zugriff: 05. Mai 2022
- [Ionos 2019] IONOS: *Reguläre Ausdrücke: Der einfache Weg, Zeichenfolgen zu beschreiben*. <https://www.ionos.de/digitalguide/websites/webseiten-erstellen/regulaere-ausdruecke/>. Version: 2019. – Letzter Zugriff: 19. April 2022

Literaturverzeichnis

- [Lehmann 2022] LEHMANN, Anne: *TTDSG-Wirrwarr: Wann (k)eine Einwilligung für Cookies erforderlich ist.* <https://legal.trustedshops.com/blog/ttdsg-wirrwarr-wann-k-eine-einwilligung-fuer-cookies-erforderlich-ist?> Version: 2022. – Letzter Zugriff: 05. Mai 2022
- [NYSE:SHOP 2022a] NYSE:SHOP: *Shopify Announces Fourth-Quarter and Full-Year 2019 Financial Results.* <https://www.businesswire.com/news/home/20200212005238/en/> *Shopify-Announces-Fourth-Quarter-and-Full-Year-2019-Financial-Results.* Version: 2022. – Letzter Zugriff: 7. März 2022
- [NYSE:SHOP 2022b] NYSE:SHOP: *Shopify Announces Fourth-Quarter and Full-Year 2020 Financial Results.* <https://www.businesswire.com/news/home/20210217005239/en/> *Shopify-Announces-Fourth-Quarter-and-Full-Year-2020-Financial-Results.* Version: 2022. – Letzter Zugriff: 7. März 2022
- [Rancea 2022] RANCEA, Bogdan: *2022 Shopify Umsatz-, Statistik- und Nutzungs- zahlen.* <https://ecommerce-platforms.com/de/articles/shopify-revenue>. Version: 2022. – Letzter Zugriff: 7. März 2022
- [Rani 2014] RANI, D. und R.: *A comparative study of SaaS, PaaS and IaaS in cloud computing.* International Journal of Advanced Research in Computer Science and Software Engineering, 2014. – Letzter Zugriff: 5. März 2022
- [Rauner 2021] RAUNER, Johannes: *TTDSG – Auswirkungen im Online Marketing.* <https://www.projecter.de/blog/affiliate-marketing/ttdsg-auswirkungen-im-online-marketing/>. Version: 2021. – Letzter Zugriff: 05. Mai 2022
- [Selenium 2022] SELENIUM: *Selenium History.* <https://www.selenium.dev/history/>. Version: 2022. – Letzter Zugriff: 10. März 2022
- [Shopify 2022] SHOPIFY: *Connect your third-party domain to Shopify manually.* <https://help.shopify.com/en/manual/domains/add-a-domain/connecting-domains/connect-domain-manual#:~:text=the%20Shopify%20IP%20address%2023.227.38.65>. Version: 2022. – Letzter Zugriff: 8. März 2022
- [shopify 2022] SHOPIFY: *Shopify-About-Homepage.* <https://www.shopify.com/about>. Version: 2022. – Letzter Zugriff: 7. März 2022
- [shopware 2022] SHOPWARE: *Shopware-Homepage.* <https://www.shopware.com/en/>. Version: 2022. – Letzter Zugriff: 5. März 2022
- [spryker.com 2022] SPRYKER.COM: *Abbildung SaaS vs. PaaS vs. On Premises.* Version: 2022. <https://spryker.com/en/paas/>. 2022. – Letzter Zugriff: 7. März 2022
- [SprykerSystems 2022] SPRYKERSYSTEMS: *Spryker-Homepage.* <https://spryker.com/en/>. Version: 2022. – Letzter Zugriff: 5. März 2022

Literaturverzeichnis

- [Usercentrics 2019] USERCENTRICS: *Cookie Banner*. <https://usercentrics.com/de/knowledge-hub/cookie-banner/>. Version: 2019. – Letzter Zugriff: 05. Mai 2022
- [Wischmeyer 2018] WISCHMEYER, Nils: *Wie Anwälte mit Abmahnungen abzocken*. <https://www.sueddeutsche.de/digital/abmahnungen-onlineshops-1.4010771>. Version: 2018. – Letzter Zugriff: 07. Mai 2022
- [Wolford 2022] WOLFORD, Ben: *What is GDPR, the EU's new data protection law?* <https://gdpr.eu/what-is-gdpr/>. Version: 2022. – Letzter Zugriff: 12. März 2022

Anhang

Aufbau einer Speicherdatei

```
1 Website 348: http://www.fairjeans.de/ Created: 08.02.2022 23:24:58
2   ↳ HEADLESS-MODE: OFF
3 [Initial] Website uses a cookie banner: True
4 [Initial] Website uses a GDPR compliant cookie banner: True
5 [Initial] Website uses a "Accept"-Button: True
6 [Initial] Website uses a "Decline"-Button (GDPR-Banner): False
7 [Initial] -> GDPR-Banner often use settings buttons.
8 #####
9 [Before] Website-Cookie-Amount: 27
10 [Before] Website-Cookies: [{"domain": "fairjeans.de", "expiry":
11   ↳ 1707431098, "httpOnly": False, ...}]
12 [Before] Cookie-Name 1: _sp_id.6260
13 [Before] Cookie-Name 2: optiMonkClientId
14 [Before] Cookie-Name 3: swym-cu_ct
15 [Before] Cookie-Name 4: _shopify_s
16 [Before] Cookie-Name 5: _shopify_y
17 [Before] Cookie-Name 6: _y
18 [Before] Cookie-Name 7: _shopify_m
19 [Before] Cookie-Name 8: swym-session-id
20 [Before] Cookie-Name 9: po_visitor
21 [Before] Cookie-Name 10: swym-o_s
22 [Before] Cookie-Name 11: _gat_gtag_UA_49552222_1
23 [Before] Cookie-Name 12: _orig_referrer
24 [Before] Cookie-Name 13: _gcl_au
25 [Before] Cookie-Name 14: swym-pid
26 [Before] Cookie-Name 15: _shopify_tw
27 [Before] Cookie-Name 16: _shopify_tm
28 [Before] Cookie-Name 17: _s
29 [Before] Cookie-Name 18: _landing_page
30 [Before] Cookie-Name 19: localization
31 [Before] Cookie-Name 20: cookieconsent_preferences_disabled
32 [Before] Cookie-Name 21: swym-swymRegid
33 [Before] Cookie-Name 22: dynamic_checkout_shown_on_cart
34 [Before] Cookie-Name 23: cart_currency
35 [Before] Cookie-Name 24: _tracking_consent
36 [Before] Cookie-Name 25: swym-email
```

Anhang

```
36 [Before] Cookie-Name 26: _sp_ses.6260
37 [Before] Cookie-Name 27: secure_customer_sig
38
39 ##### After accepting the Cookie-Banner (or not) #####
40 [After] Website-Cookie-Amount: 35
41 [After] Website-Cookies: [{domain: '.fairjeans.de', 'expiry':
42   ↳ 1644359165, 'httpOnly': False, ...}]
43 [After] Cookie-Name 1: _gat
44 [After] Cookie-Name 2: _gid
45 [After] Cookie-Name 3: _tracking_consent
46 [After] Cookie-Name 4: cart_sig
47 [After] Cookie-Name 5: cart
48 [After] Cookie-Name 6: _sp_id.6260
49 [After] Cookie-Name 7: cart_ts
50 [After] Cookie-Name 8: cookieconsent_preferences_disabled
51 [After] Cookie-Name 9: dynamic_checkout_shown_on_cart
52 [After] Cookie-Name 10: swym-swymRegid
53 [After] Cookie-Name 11: optiMonkClientId
54 [After] Cookie-Name 12: swym-cu_ct
55 [After] Cookie-Name 13: _shopify_s
56 [After] Cookie-Name 14: cart_ver
57 [After] Cookie-Name 15: _shopify_y
58 [After] Cookie-Name 16: _y
59 [After] Cookie-Name 17: _ga
60 [After] Cookie-Name 18: _shopify_m
61 [After] Cookie-Name 19: swym-session-id
62 [After] Cookie-Name 20: po_visitor
63 [After] Cookie-Name 21: swym-o_s
64 [After] Cookie-Name 22: _gat_gtag_UA_49552222_1
65 [After] Cookie-Name 23: _orig_referrer
66 [After] Cookie-Name 24: _gcl_au
67 [After] Cookie-Name 25: swym-pid
68 [After] Cookie-Name 26: _shopify_tw
69 [After] Cookie-Name 27: cookieconsent_status
70 [After] Cookie-Name 28: _shopify_tm
71 [After] Cookie-Name 29: _s
72 [After] Cookie-Name 30: _landing_page
73 [After] Cookie-Name 31: localization
74 [After] Cookie-Name 32: _sp_ses.6260
75 [After] Cookie-Name 33: swym-email
76 [After] Cookie-Name 34: cart_currency
77 [After] Cookie-Name 35: secure_customer_sig
78 ###### Third party cookies used before acceptance #####
79 [Used before] Cookie-Name: _gcl_au (Google Adsense)
80
```

Anhang

```
81 ##### Difference (Before/After) #####
82 [Difference] Cookies added: 8
83 [Difference] Cookie-Name: 1: _gid
84 [Difference] Cookie-Name: 2: cart_ver
85 [Difference] Cookie-Name: 3: cart_ts
86 [Difference] Cookie-Name: 4: cart
87 [Difference] Cookie-Name: 5: _ga
88 [Difference] Cookie-Name: 6: cookieconsent_status
89 [Difference] Cookie-Name: 7: _gat
90 [Difference] Cookie-Name: 8: cart_sig
91
92 ##### Analysis #####
93 [Analysis] Website offers the user, in addition to accepting the
94   ↳ cookie-banner, the possibility to reject it (GDPR): False
94 [Analysis] Website generally uses cookies at the beginning
94   ↳ (non-essential), although they are not authorized by the user:
94   ↳ True
95 [Analysis] Website uses unauthorized third-party cookies at the
95   ↳ beginning: True
96 [Analysis] Website respects the user's decision and loads
96   ↳ THIRD-PARTY-COOKIES only after approval: False
97
98 [Analysis] Website is NOT GDPR compliant!
```

Metadaten-Auflistung

Metadatum 1:

Name des Online-Shops, Erstellungsdatum, Headless-Mode (AN/AUS)(im späteren Verlauf näher erläutert)

Erläuterung:

Dient der einfachen Wiedererkennung einer Speicherdatei. Welche Speicherdatei gehört zu welchem Online-Shop

Bedeutung/Einfluss bzgl. der Entscheidung:

-

Anhang

Metadatum 2:

Online-Shop verwendet einen Cookie-Banner (True/False)

Erläuterung:

Gibt an, ob ein Online-Shop allgemein einen Cookie-Banner verwendet und diesen dem Nutzer auch anzeigt

Bedeutung/Einfluss bzgl. der Entscheidung:

Hat keinen direkten Einfluss auf die Entscheidung. Beeinflusst allerdings die anderen Metadaten. Sobald dieses Feld False ist, kann die Berechnung der anderen Metadaten aus Laufzeit-Gründen übersprungen werden

Metadatum 3:

Online-Shop verwendet einen DSGVO-konformen Cookie-Banner (True/False)

Erläuterung:

Gibt an, ob ein Online-Shop einen DSGVO-konformen Cookie-Banner verwendet und diesen dem Nutzer auch anzeigt

Bedeutung/Einfluss bzgl. der Entscheidung:

Hat einen direkten Einfluss auf die Entscheidung, ob ein Online-Shop DSGVO-konform ist oder nicht. Bei False ist bereits die Entscheidung final als nicht DSGVO-konform eingestuft

Metadatum 4:

Online-Shop verwendet einen „Akzeptieren“-Knopf (True/False)

Erläuterung:

Gibt an, ob ein Cookie-Banner einen „Akzeptieren“-Knopf besitzt und dieser auch vom Nutzer betätigt werden kann

Bedeutung/Einfluss bzgl. der Entscheidung:

Dient der allgemeinen Analyse des Shop-Verhaltens

Metadatum 5:

Online-Shop verwendet einen „Ablehnen“-Knopf (True/False)

Erläuterung:

Gibt an, ob ein Cookie-Banner einen „Ablehnen“-Knopf besitzt und dieser auch vom Nutzer betätigt werden kann

Bedeutung/Einfluss bzgl. der Entscheidung:

Dient der allgemeinen Analyse des Shop-Verhaltens

Anhang

Metadatum 6:

Online-Shop verwendet DSGVO-konforme Auswahlmechanismen für den Nutzer (Wird nur erfasst, wenn ein DSGVO-konformer Cookie-Banner vorhanden ist und angezeigt wird) (True/False)

Erläuterung:

Gibt an, ob ein Cookie-Banner verschiedene DSGVO-konforme Auswahlmechanismen besitzt und diese auch dem Nutzer anzeigt

Bedeutung/Einfluss bzgl. der Entscheidung:

Hat einen direkten Einfluss auf die Entscheidung, ob ein Online-Shop DSGVO-konform ist oder nicht. Bei False ist bereits die Entscheidung final als nicht DSGVO-konform eingestuft

Metadatum 7:

Cookie-Unterschied (Davor/Danach)

Erläuterung:

Eine Auflistung der Cookies, welche nach der Akzeptierung des Cookie-Banners dazugekommen sind

Bedeutung/Einfluss bzgl. der Entscheidung:

Wird unter anderem für einzelne Berechnungen als Parameter verwendet. „Auffällig“ sind hierbei Online-Shops, welche nach der Akzeptierung eine größere Menge an Cookies erzeugen. Zudem wird dieses Metadatum ebenfalls dazu verwendet, um das allgemeine Online-Shop Verhalten zu dokumentieren

Metadatum 8:

Besondere Merkmale (True/False)

Erläuterung:

In manchen Fällen besitzen Online-Shops Charakteristiken für einen Cookie-Banner. Dieser wird allerdings nicht dem Nutzer angezeigt

Bedeutung/Einfluss bzgl. der Entscheidung:

Hat einen direkten Einfluss auf die Entscheidung, ob ein Online-Shop DSGVO-konform ist oder nicht. Bei True ist bereits die Entscheidung final als nicht DSGVO-konform eingestuft

Domain-Text-Datei-Auflistung

```
99 http://levlup.de/
100 http://gladskin.de/
101 http://craemerco.de/
102 http://malerfachhandel24.de/
103 http://thingo24.de/
104 http://patangroup.de/
105 http://nordgreen.de/
106 http://adoreit.de/
107 http://spacedoutclo.de/
108 http://halter-edelbraende.de/
109 http://gitticonsciousbeauty.de/
110 http://panellia.de/
111 http://sakerwerkzeuge.de/
112 http://pumpkin-organics.de/
113 http://stupidfish.design/
114 http://dierollerscheune.de/
115 http://airfield.de/
116 http://boogybooty.de/
117 http://duschhaus.de/
118 http://flederrausch.de/
119 http://hamburg-zanzibar.de/
120 http://ivy-oak.de/
121 http://jesango.de/
122 http://jeunepremier.de/
123 http://junoandme.de/
124 http://leif-nelson.de/
125 http://micaraa.de/
126 http://nu3.de/
127 http://oneodio.de/
128 http://dogslover.de/
129 http://vejo.de/
130 http://chessbazaar.de/
131 http://nd24.de/
132 http://zauber-puzzle.de/
133 http://charlesandmarie.de/
134 http://haustierdreams.de/
135 http://mammaly.de/
136 http://paexfood.de/
137 http://biocbd.de/
138 http://everdrop.de/
139 http://viral-protect.de/
140 http://vitalbodyplus.de/
141 http://salnatural-shop.de/
142 http://ooono.de/
```

Anhang

```
143 http://erfindar.de/  
144 http://reternity.de/  
145 http://ichbinpur.de/  
146 http://get-icy.de/  
147 http://organisation.de/  
148 http://soraskystore.de/  
149 [+3.028 weitere]
```

Konsolenausgabe „ExtractData“-Skript (deutsche Ausgabe)

```
150 ##### Grundlegende Ergebnisse #####  
151 Anzahl der Webseiten, welche einen cookie banner verwenden (Merkmale  
  ↳ besitzen): 2399 => [Prozentsatz: 99.05%], [23 Webseite(n) nicht]  
152 Anzahl der Webseiten, welche einen GDPR konformen cookie banner  
  ↳ verwenden: 1578 => [Prozentsatz: 65.15%], [844 Webseite(n) nicht]  
153 Anzahl der Webseiten, welche einen "Akzeptieren"-Button (Merkmale  
  ↳ davon) verwenden: 2399 => [Prozentsatz: 99.05%], [23 Webseite(n)  
  ↳ nicht]  
154 Anzahl der Webseiten, welche einen "Ablehnen"-Button (Merkmale davon)  
  ↳ verwenden: 29 => [Prozentsatz: 1.2%], [2393 Webseite(n) nicht]  
155 Anzahl der Webseiten, welche GDPR-konforme Auswahlmechanismen  
  ↳ anbieten: 528 => [Prozentsatz: 21.8%], [1894 Webseite(n) nicht]  
156 ##### Cookie-Amount-Ergebnisse #####  
157 ---Before:  
158 Anzahl der Webseiten, welche weniger als 10 cookies vor der  
  ↳ Akzeptierung verwendet haben: 493 [20.36%]  
159 Anzahl der Webseiten, welche zwischen 10 und 20 cookies vor der  
  ↳ Akzeptierung verwendet haben: 1177 [48.6%]  
160 Anzahl der Webseiten, welche zwischen 20 und 30 cookies vor der  
  ↳ Akzeptierung verwendet haben: 661 [27.29%]  
161 Anzahl der Webseiten, welche zwischen 30 und 40 cookies vor der  
  ↳ Akzeptierung verwendet haben: 83 [3.43%]  
162 Anzahl der Webseiten, welche zwischen 40 und 50 cookies vor der  
  ↳ Akzeptierung verwendet haben: 8 [0.33%]  
163 Anzahl der Webseiten, welche mehr als 50 cookies vor der Akzeptierung  
  ↳ verwendet haben: 0 [0.0%]  
164 ---After:  
165 Anzahl der Webseiten, welche weniger als 10 cookies nach der  
  ↳ Akzeptierung verwendet haben: 126 [5.2%]  
166 Anzahl der Webseiten, welche zwischen 10 und 20 cookies nach der  
  ↳ Akzeptierung verwendet haben: 1048 [43.27%]  
167 Anzahl der Webseiten, welche zwischen 20 und 30 cookies nach der  
  ↳ Akzeptierung verwendet haben: 858 [35.43%]
```

Anhang

169 Anzahl der Webseiten, welche zwischen 30 und 40 cookies nach der
 ↳ Akzeptierung verwendet haben: 343 [14.16%]
170 Anzahl der Webseiten, welche zwischen 40 und 50 cookies nach der
 ↳ Akzeptierung verwendet haben: 47 [1.94%]
171 Anzahl der Webseiten, welche mehr als 50 cookies nach der Akzeptierung
 ↳ verwendet haben: 0 [0.0%]
172 ---Difference:
173 Unterschied cookie-anzahl (davor/danach), weniger als 10 cookies
 ↳ dazugekommen: 2012 [83.07%]
174 Unterschied cookie-anzahl (davor/danach), zwischen 10 und 20 cookies
 ↳ dazugekommen: 310 [12.8%]
175 Unterschied cookie-anzahl (davor/danach), zwischen 20 und 30 cookies
 ↳ dazugekommen: 96 [3.96%]
176 Unterschied cookie-anzahl (davor/danach), zwischen 30 und 40 cookies
 ↳ dazugekommen: 4 [0.17%]
177 Unterschied cookie-anzahl (davor/danach), zwischen 40 und 50 cookies
 ↳ dazugekommen: 0 [0.0%]
178 Unterschied cookie-anzahl (davor/danach), mehr als 50 cookies
 ↳ dazugekommen: 0 [0.0%]
179 ##### Third-Party-Cookie-Ergebnisse #####
180 Anzahl der Webseiten, welche den facebook-pixel als third-party-cookie
 ↳ unerlaubt verwendet haben: 913 => [Prozentsatz: 37.7%], [1509
 ↳ Webseite(n) nicht]
181 Anzahl der Webseiten, welche den pinterest-tag als third-party-cookie
 ↳ unerlaubt verwendet haben: 294 => [Prozentsatz: 12.14%], [2128
 ↳ Webseite(n) nicht]
182 Anzahl der Webseiten, welche einen google service unerlaubt als
 ↳ third-party-cookie verwendet haben: 1194 => [Prozentsatz: 49.3%],
 ↳ [1228 Webseite(n) nicht]
183 Anzahl der Webseiten, welche hotjar als third-party-cookie unerlaubt
 ↳ verwendet haben: 175 => [Prozentsatz: 7.23%], [2247 Webseite(n)
 ↳ nicht]
184 Anzahl der Webseiten, welche hubspot als third-party-cookie unerlaubt
 ↳ verwendet haben: 25 => [Prozentsatz: 1.03%], [2397 Webseite(n)
 ↳ nicht]
185 Anzahl der Webseiten, welche klaviyo als third-party-cookie unerlaubt
 ↳ verwendet haben: 477 => [Prozentsatz: 19.69%], [1945 Webseite(n)
 ↳ nicht]
186 Anzahl der Webseiten, welche leadfeeder als third-party-cookie
 ↳ unerlaubt verwendet haben: 2 => [Prozentsatz: 0.08%], [2420
 ↳ Webseite(n) nicht]
187 Anzahl der Webseiten, welche microsoft clarity als third-party-cookie
 ↳ unerlaubt verwendet haben: 80 => [Prozentsatz: 3.3%], [2342
 ↳ Webseite(n) nicht]

Anhang

```
189 Anzahl der Webseiten, welche lucky orange als third-party-cookie
    ↳ unerlaubt verwendet haben: 14 => [Prozentsatz: 0.58%], [2408
    ↳ Webseite(n) nicht]

190
191 ##### Finale Analyse #####
192 Anzahl der Webseiten, welche cookie-merkmale besitzen aber dem Nutzer
    ↳ keinen cookie-banner anzeigen (Sondermerkmal): 953 =>
    ↳ [Prozentsatz: 39.35%], [1469 Webseite(n) nicht]

193 Anzahl der Webseiten, welche dem Nutzer eine "Ablehn"-Option anbieten:
    ↳ 521 => [Prozentsatz: 21.51%], [1901 Webseite(n) nicht]

194 Anzahl der Webseiten, welche third-party-cookies am Anfang unerlaubt
    ↳ nutzen: 1417 => [Prozentsatz: 58.51%], [1005 Webseite(n) nicht]

195 Anzahl der Webseiten, welche die Entscheidung des Nutzers respektieren
    ↳ und third-party-cookies nur dann laden, wenn dies ausdrücklich
    ↳ gewünscht ist: 973 => [Prozentsatz: 40.17%], [1449 Webseite(n)
    ↳ nicht]

196
197 Anzahl der Webseiten, welche GDPR-konform sind: 743 => [Prozentsatz:
    ↳ 30.68%], [1679 Webseite(n) nicht]
```

Github-Repository

Github-Repository des Projekts: <https://github.com/MHallweger/GDPR-Cookie-Crawler>

Eidesstattliche Erklärung

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben.

Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Halver, 11. Mai 2022



Marvin Nicholas Hallweger

