Politecnico Di Torino

Master's Degree in Computer Engineering

AwEE the AwarE Extension - Enhancing Privacy Awareness by Leveraging Gamification Principles

Politecnico di Torino

Supervisors
Antonio Vetrò
Riccardo Coppola
Lorenzo Laudadio

Candidate
Luca Calò

Academic Year 2024/2025

Summary

In today's digital society, companies regularly collect large volumes of personal data from their users, raising serious privacy and ethical concerns. Yet, despite many cases of large data breaches being documented, most users remain unaware of the real-time flow of their information while they use online services.

This thesis presents the design and development of AwEE, a software conceived to address this gap by improving privacy awareness among users, incorporating gamification principles.

While users navigate a given website through their browser, various HTTP requests are made. Among these, a considerable number are directed to entities that are not directly related to the site visited (e.g., telemetry, profiling for advertising purposes), in a manner that is completely opaque to the user. AwEE's goal is precisely to reveal to users that, while they are browsing, they are also sending some of their personal data[1] to third parties. Knowing this, they may make different choices when browsing than they would if they were unaware of what is happening.

The first step towards the creation of AwEE was studying privacy-related research literature which revealed a number of existing software programs. Some of these (FPMON, FPNET, Minos) aim to improve users' perception of how they are profiled while browsing; others (JShelter, NoScript) aim to enhance users' privacy during navigation, trying to prevent profiling from happening.

Then, attention turned to the concept of "gamification", meaning the use of game design mechanics in non-gaming contexts, and examples of its applications. The main reference point from this perspective was the Octalysis Framework, a well known scheme defining fundamental psychological motivators that drive human actions.

Bearing this background in mind, AwEE was developed as a browser extension, available for both Firefox and Chromium. It allows users to monitor outgoing HTTP requests

[1]Referring to the GDPR definition of "personal data", i.e., any type of information that can lead to the identification of a natural person. Since when a device makes an HTTP request to a certain server, the server can identify the device from its IP address, it follows that even the IP address of a host is a personal data, and each HTTP request is a personal data transfer.

while browsing the web and flags those directed to "Bad Hosts" – companies fined or under investigation for questionable data practices – using an internal blacklist to maintain a serverless approach.

To increase users' engagement, various gamification elements were integrated. Among these, the tool includes the possibility for users to choose their own personal avatar, which also allows them to receive dynamic feedback as they navigate, simultaneously stimulating a sense of ownership and unpredictability; meanwhile the graphical interface includes counters and statistics, which involve users by updating them in real-time on what is happening while they browse.

A preliminary evaluation of AwEE was conducted in an academic context involving a total of 120 participants among high school and bachelor students. The outcome of such assessment highlights both extension's usability and ability to shift users' perceptions about data transfers to big tech companies, demonstrating the potential of combining privacy tools with gamification.

Possible technical improvement would be to explore alternative methods to automatize the recognition of "Bad Hosts", evaluate the introduction of a server-based backend, and conduct more in-depth experiments in order to better understand AwEE's capabilities. Additional and more complex gamification mechanics could also be implemented, allowing AwEE to evolve as a powerful tool for promoting privacy awareness while keeping users motivated and engaged in safeguarding their personal data.

Contents

"A convivial society should be designed to allow all its members the most autonomous action by means of tools least controlled by others."

- Ivan Illich, Tools of Conviviality (1973)

Chapter 1

Introduction

1.1 The age of data extraction

Today's world and society are strongly characterized by the ubiquitous pervasiveness of digital connections and the continuous extraction of value from every human activity in the form of data.

There are now many companies that have built or adapted their business model around this extractive possibility, guaranteed by the increasing diffusion and demand for digital commodities and personalized services.

Shoshana Zuboff defines the above picture as "Surveillance Capitalism". This is a new form of capitalism that claims human experience as free raw material translated into data and used to manufacture predictive products that anticipate people's wants or needs. Ultimately, this provides powerful corporations the ability to manipulate and control our behaviors [1].

Nick Couldry and Ulises A. Mejias, on the other hand, emphasize how, in extracting data from people, infrastructures and extraction practices similar to historical colonialism are used. They define this process as "Data Colonialism", identifying it as a key dimension of the current phase of capitalism's expansion. In this way, they reveal themselves to be more connected to the Marxian tradition and not to believe in the realization of new forms of capitalism[2].

Regardless of the different theoretical nuances one decides to give to the current state of affairs, it is evident that this continuous flow of data and information creates concerns about the users' privacy and how ethically their data are handled, while there are various cases that shifted the public debate on these issues.

1.2 Cases of Concern

Several high-profile cases have highlighted the potential for abuse of the aforementioned data mining practices. From government agencies to private companies, the collection and exploitation of personal data have sparked heated debates about the balance between security and individual rights. The following examples illustrate the scope and complexity of digital surveillance and data exploitation, and the significant consequences for individuals and communities. Precisely for this reason, they highlight the need for transparency, accountability, and significantly better safeguards.

In 2013, Edward Snowden, a former CIA technician and contractor for the U.S. National Security Agency (NSA), disclosed a trove of classified documents revealing extensive global surveillance programs run by the NSA and its international partners. Leaks revealed that the agency collected telephone metadata in bulk from major telecommunications companies and had direct access to the servers of major internet providers [3].

Similarly, the Cambridge Analytica scandal in 2018 showed how millions of Facebook users' personal data was obtained without their authorization to craft targeted political ads during the 2016 US presidential election campaign [4].

Later, a lawsuit started in 2020, forced Google to agree to a $5 billion settlement in 2024, destroy billions of records and update its data collection practices. It alleged that the company secretly tracked users as they browsed in incognito mode, turning it into a "mine of unaccountable information" [5].

In 2022, researcher Felix Krause demonstrated that the browser built into the TikTok app injects JavaScript that can capture every keystroke and other interactions on third-party web pages opened within the app [6].

More recently, in the context of the Israeli-Palestinian conflict that reignited following the October 7 attacks [7], hypothesized privacy threats found dramatic concrete evidence: continuous data-extraction systems were adopted on Palestinian population.

In 2023, Israel's systematic use of biometric surveillance infrastructure targeting Palestinian civilians and the creation of "Lavender", an AI targeting system with little human oversight and a permissive casualty policy used by the Israeli army were demonstrated [8] [9].

Then, in 2025, Israeli intelligence deployed an LLM using millions of intercepted conversations between Palestinians, with the objective of speeding up the indictment and arrest process [10].

It should be noted that these reported cases are only a few examples, but they help the reader to understand the magnitude of the dynamics described and their influence on today's society.

1.3 GDPR

The above concerns led to multiple attempts to legislate against the indiscriminate harvesting and usage of personal data, in order to ensure the protection of users' privacy.

In this regard the European Union's General Data Protection Regulation (GDPR) [11] is the most significant and emblematic example of a unified framework to give citizens more control over their data. It is a law that establishes how organizations and companies must behave when processing personal data in any way, ensuring that they act in full compliance with privacy regulations. In the context of the GDPR, "personal data" refers to any type of information that can directly or indirectly identify an individual.

More specifically, but without claiming to be exhaustive, it establishes that every organization dealing with personal data is required to always specify what data it collects and for what purposes, prohibiting the gathering of data in advance for unspecified future use and forcing for their deletion when the purpose for which they were collected for has been fulfilled. Furthermore, data collection, processing, and storage must be carried out securely, while ensuring that individuals can access, request to modify, or request the deletion of their data at any time.

The GDPR is a perfect example of the "Brussels Effect", i.e., the process that allows laws made in the European Union to cross its borders, influencing the legislation of other states [12]. It has become a reference model around the world, and several countries have adopted laws directly inspired by European legislation over the years, including Brazil [13], Canada [14], China [15], and some U.S. States like California [16], Virginia [17], and Colorado [18].

However, the introduction of cutting-edge laws is only part of the changes needed for user privacy to be truly protected. It is necessary, above all, to change the habits of people, who got used to surrendering personal information in exchange for seemingly free (in terms of money) services.

1.4 Thesis proposal

The aim of this thesis project is to present and describe AwEE, a new browser extension developed to increase users' awareness of personal data transfer tracking during web browsing, harnessing a gamified approach.

The initial focus was on studying existing literature on privacy-related research. These studies revealed a number of software programs with the aim of protecting privacy or raising user awareness in this regard, the main features of which are described further on, in 2.1.

Once a general background on existing projects focusing on privacy protection had been established, attention turned to understanding the concept of "gamification", meaning the use of game design mechanics in non-gaming contexts, and to examples of its applications (2.2).

At this point, after gaining an overview of the general structure of extensions for the most common browsers, the design and subsequent development of the software began.

The tool works by making online data flows visible and understandable as they occur. It monitors outgoing HTTP requests while browsing the web and highlights *Bad Requests* to domains associated with *Bad Hosts*, defined as companies that have been fined or are under investigation for problematic data practices. To maintain a serverless approach, these *Bad Hosts* are contained in an internal *blacklist*, within a JSON file. Both design and implementation choices have been discussed in detail in 3.

Later on, in 5, the brief experiment conducted to obtain a preliminary evaluation of AwEE is described, and its results are presented.

In 4, a list of privacy-related browser extensions that had an impact on the AwEE development process is provided.

In 6, possible threats to the validity of the work are stated, and finally in 7, main contributions are summarized and an overview of possible future improvements and research directions are indicated.

Chapter 2

Background

The purpose of this chapter is to describe the context in which AwEE was developed, specifying which works and projects inspired and influenced its implementation, considering aspects related to privacy 2.1 and gamification 2.2.

2.1 Privacy Related Studies

The main motivation behind the development of AwEE is the growing amount of concerns about online privacy and the increasing spread of data-tracking policies adopted by big tech companies. This led to a growing need for tools that empower users to make informed decisions about their privacy and manage their personal information more consciously. Several studies in the literature address privacy protection on the web.

2.1.1 Some remarkable examples

Garimella et al. [19] measured the impact of widely used ad-blocking extensions on page load performance, assessing how websites adapt to users adopting these tools and how they react when they detect blocking. They then investigated the ad-blockers capability to prevent the transfer of user tracking information, resulting in privacy benefits for people.

Other studies focused on the creation of new tools, such as FPMON [20] and FPNET [21], designed to measure and rate fingerprinting activity on websites. Fingerprinting is a technique that uniquely identifies users based on their browser and hardware device characteristics, allowing for pervasive tracking even without the use of cookies. Specifically, FPMON is a browser extension that allows users to visualize a measurement and

evaluation of the fingerprinting activity of the website they are visiting, in real time. FP-NET, on the other hand, is an automated scalable and robust tool based on FPMON, able to analyze a large set of websites and detect fingerprinting networks by observing their behavior.

Another example is JShelter [22], which is a browser extension that allows users to tweak and limit the action of web browser APIs, prevent fingerprinting, and avoid attacks that exploit information about the device, browser, user, and location.

Lastly, NoScript [23] is another browser extension whose main feature is to allow the execution of JavaScript and other potentially unsafe content only on websites that the user considers reliable, giving them the option to define these trustworthy sites through a whitelisting mechanism. In addition to this core functionality, NoScript also provides some additional protections, the most important of which is an XSS filter, which prevents requests from a given website from injecting and running code on another site, resulting in a Cross Site Scripting (XSS) attack [24].

2.1.2 Minos

The main starting point for the development of AwEE was definitely *Minos*, a user-friendly application developed as an Electron App by Lorenzo Laudadio *et al.* [25][26] that allows to browse the Web while logging HTTP requests. The software combines a simple browser with a backend tailored to record and analyze personal data transfers[1] to countries outside the European Economic Area (EEA). Whenever a personal device makes an HTTP request to a certain server, the server can identify the device from its IP address. It follows that even the IP address of a host can be considered as personal data, and each HTTP request results in a personal data transfer.

In this context, the focus of *Minos* shifts to verifying, among all HTTP requests, which ones are directed to non-EEA countries, seeking to detect potential violations of the GDPR provisions. In fact, it states that transfers of personal data can only be made to countries that are part of the EU, EEA countries, countries that have obtained an "adequacy decision" from the European Commission (an official statement that the particular country provides adequate protection for personal data), or even countries that, in the absence of the "adequacy decision", nonetheless provide adequate additional safeguards for data protection.

Moreover, the application was used to analyze a certain number of Italian Public Administration websites.

[1] Referring to the GDPR definition of "personal data", i.e., any type of information that can lead to the identification of a natural person. See 1.3.

For what concerns AwEE, all this attention given to the geographic location of the domains to which requests are made is not directly interesting, since within it, the focus is on the companies which are behind such domains. Specifically, big tech companies or entities that suffered fines or are under investigation for their bad data handling practices are taken into account.

2.2 Octalysis Framework

One of the main objectives of the thesis project was to effectively exploit gamification principles within the extension. In order to do so, it was necessary to study the existing literature on the subject in depth.

Citing Deterding et al., gamification can be defined as the adoption of mechanics typical of game design in a non-ludic context [27].

Many examples of works in literature focus on the impact of gamification in privacy protection. *Mavroeidi et al.* explored the existing connections between privacy and gamification, highlighting the need for programs that educate users on privacy by adopting gamification mechanics [28]. The same authors also explored how gamification can be used in designing Privacy Training Programs [29], while *Ruggiu et al.* highlighted the significance of paying attention to privacy when applying gamification to work environments [30].

The design of the gamification mechanics of AwEE is based on the *Octalysis Framework*, developed by Yu-kai Chou [31].

First of all, he actually prefers to redefine the term "gamification" as *Human-Focused Design*, in opposition to what is normally found in society, which is *Function-Focused Design*. While the latter focuses on getting the job done quickly, viewing people as cogs in a system, *Human-Focused Design* considers the feelings, motivations, ambitions, uncertainties, and engagement that leads a person to want to perform actions.

The reason why *Human-Focused Design* took the name "gamification", is because the first industry capable of developing skills in this field was the gaming industry, where game-designers spent decades figuring out how to keep their customers continuously engaged.

Yu-Kai Chou then defines two ways of implementing gamification.

The first is explicit gamification, which uses actual games to achieve non-game goals. This certainly guarantees the creation of a more playful environment, while at the same time giving designers great creative freedom, since it explicitly involves proper games.

The downsides of doing this are the possibility of not being taken seriously and that often it requires a greater amount of resources.

The second, on the other hand, is implicit gamification, which instead consists of a form of design that subtly exploits the elements and techniques typical of games, inserting them as invisibly as possible into the user experience. This approach is technically easier to implement, and in most cases it is the most appropriate, but the fact that it is so convenient can often lead to poor and lazy designs if the necessary precautions are not taken.

The framework thought up by Yu-Kai Chou is called *Octalysis* because it can be represented as an octagonal-shape scheme (see Figure 2.1) and it's based on the eight *Core Drives*. They are defined as the fundamental psychological motivators that drive every human action.

Below is a brief summary of these *Core Drives*.

2.2.1 Core Drive 1: Epic Meaning & Calling

Epic Meaning & Calling is the *Core Drive* in play when a person believes that they are doing something greater than themselves, that they have been chosen for taking that action.

Many games exploit this Core Drive, for example by creating a fantasy narrative in which the world is about to be destroyed and the player is the only one who can save it, or by telling them that they are a member of an organization and must do their part to accomplish a greater mission.

However, this type of motivation can also be applied to everyday activities, regardless of the explicit context of games.

An example of this is the free online encyclopedia Wikipedia, hosted since 2003 by the non-profit organization Wikimedia Foundation [32], maintained and managed entirely by a community of volunteers who are committed to democratizing and preserving knowledge in the world.

Or, moving on to a more social and cultural dimension, Yu-Kai Chou cites the way parenthood is conceived in Chinese tradition: children are indebted to their parents from the moment they are born because they owe them their lives and, because of this, they feel obliged to honor and support them in all circumstances. This concept contrasts with what happens in Western culture, where parents tend to motivate their children by rewarding them when they behave well and punishing them when they behave badly, exploiting *Core Drive 2* and *Core Drive 8*.

2.2.2 Core Drive 2: Development & Accomplishment

Development & Accomplishment is the *Core Drive* for making progress, developing skills, achieving mastery, overcoming challenges. It has to do, for example, with focusing on career goals, with the motivation generated by acquiring new abilities, and with showing where you have managed to get to, what you have managed to do.

This is the easiest *Core Drive* to design and implement, and in fact, many companies focus almost exclusively on this when they intend to exploit gamification.

Many products use scoring systems, badges, and rankings to highlight achievements, but, as Yu-Kai Chou points out, these often end up being somewhat self-serving exercises. When trying to leverage gamification, it is important not to think about which elements or mechanics to use, but how you want the user to feel (this applies to all *Core Drives*, not just *Development & Accomplishment*). The key to using *Core Drive 2* correctly, in fact, lies in showing them the rewards promised for overcoming any challenge. This has to be done while always ensuring that they feel proud to have reached these achievements.

To cite some uses of this type of motivation, Amazon, for example, inserted a leaderboard in its buyer review interface to express the feelings of the community through star ratings and scores. Or Ebay, which, based on an online auction system, gives the buyer not so much the idea of buying an item, but of winning it. It features a scoring system that ranks both sellers, rewarding their skills with specific badges, and buyer-seller feedback.

2.2.3 Core Drive 3: Empowerment of Creativity & Feedback

Empowerment of Creativity & Feedback is the *Core Drive* triggered when people are engaged in a creative process, they repeatedly figure new things out, try different combinations, and receive feedback about it.

In fact, it is linked to everything normally associated with the word "Play." Consequently, by unlocking it and exploiting it, it is possible to generate an evergreen engine of engagement, which easily flows into the other *Core Drives*.

Unfortunately, however, all this potential is difficult to unlock because *Core Drive 3* is also the most complicated to implement: it requires the user to devote a certain amount of attention, which is by no means a given in today's attention-deficit and information-bombing society.

As already mentioned, it is a fundamental part of actual games. For example, it is the core motivator of chess, where the practically infinite possibilities for variations in the game allow players to express their creativity and develop their own personal strategies. Something similar happens in battle card games, example of a setup in which the user is given a goal (victory) and a variety of tools to combine in order to create a strategy for achieving that goal.

An example instead of the use of this *Core Drive* embedded in the workplace is "20% Time" Google's program, that allowed employees to take one day a week to work on a personal side project [33]. This approach was able to engage employees by allowing them to be creative and provide feedback in the workplace, giving voice to their personal leanings.

2.2.4 Core Drive 4: Ownership & Possession

Ownership & Possession is the *Core Drive* in place when people feel like they own or control something, perceiving the innate motivation to increase or improve what they possess.

It is based on the connection with the investment in terms of time and resources spent on customizing something according to one's tastes. This generates commitment, as well as a need for consistency with previous choices.

It is a key motivator in scenarios where elements such as virtual currencies or goods are used, as well as in the context of collectibles research activities.

A representative example of this *Core Drive* application is undoubtedly Tamagotchi, a life simulator whose purpose was to care for a small animal, providing it with the necessities of life and giving it the appropriate education [34]. Its great success proved that when people feel they own something, they are naturally driven to protect, take care of, and commit themselves to it.

2.2.5 Core Drive 5: Social Influence & Relatedness

Social Influence & Relatedness is the *Core Drive* that incorporates all social elements motivating people such as mentorship, acceptance, companionship, competition, envy, and includes also the sense of feeling related to something, like for example nostalgia.

It is strongly linked to the human need to connect with other people and interact with them. Nowadays, virtually every consumer service exploits this *Core Drive* by suggesting users to invite their friends to sign up for the service. However, motivating them in this manner can be a double-edged sword: if users begin to associate this type of dynamic with mere marketing, they lose trust and stop feeling involved.

There is a very strong correlation between *Core Drive 5* and *Core Drive 1*, in that when a person believes that an action adheres to social norms, they will tend to do it (*Core Drive 5*), but this also happens because such behavior makes them feel part of a larger group, potentially an elite one (*Core Drive 1*).

An example of this was the famous commercial created by the Keep America Beautiful Organization known as "Crying Indian", which depicted a Native American crying after

seeing people throw trash out of their cars [35]. It was so effective that it is considered one of the best public service announcements of all time.

Despite this success, a subsequent ad that reinterpreted it several years later, which showed a poster of the same Native American tear, but with the slogan "Back by popular neglect", was not only ineffective but even counterproductive. In fact, it showed the public that most people pollute, defining it as a social norm [36]. So although the message was intended to be about avoiding pollution, in the end the campaign actually encouraged people to be more like this negative norm.

2.2.6 Core Drive 6: Scarcity & Impatience

Scarcity & Impatience is the *Core Drive* of wanting something because it's extremely rare or exclusive, or because is not immediately obtainable.

The motivation that derives from this is closely linked to the human tendency to desire what they cannot have.

Furthermore, our perception is often influenced more by relative changes than by absolute ones, so that getting $10 million is felt very differently if I previously had $1 million or $1 billion: perceiving abundance implies a decrease in motivation.

Yu-Kai Chou cites "Geomon", created by Loki Studios as a representative example of the use of this *Core Drive* [37]. It was a game in which players tried to capture monsters in order to then fight each other. In fact, it was a precursor to Pokémon Go, as also in the case of Geomon the possibility of finding specific monsters was influenced by the physical location of the player, obtained via the phone GPS. Since some monsters were only found in specific locations, they were very rare, and for this reason many people were willing to pay real money to own them.

2.2.7 Core Drive 7: Unpredictability & Curiosity

Unpredictability & Curiosity is the *Core Drive* that brings a person to constantly be engaged not knowing what is going to happen next.

We are particularly attracted to experiences that are characterized by uncertainty and chance. We are more involved in situations that involve the possibility of winning rather than those in which we know our odds for certain. When we know we will receive a reward, the excitement reflects only the emotional value of that reward. If we only have the chance to win, we are much more involved because of the excitement of whether we will win or not.

Among the examples of how this *Core Drive* is used, we can mention the "I'm feeling lucky" button on the Google search engine home page. While normal searches on

Google.com give the user a list of web pages curated by Google that best match the keywords entered, using the "I'm Feeling Lucky" button allows bypassing this. It redirects the user immediately to the web page that would be at the top of the results page.

From the user's point of view, this approach creates an unpredictable experience, with a certain degree of suspense, creating engagement with the tool.

Or take the case of Woot!, an e-commerce site that combines *Core Drive 6* and *Core Drive 7* [38].

The main website originally offered only one discounted product per day. In fact, users never knew in advance what the product would be, and there was always a maximum limit on the number of people who could make the purchase, triggering the scarcity motivator since no one knew how long the offer would last. Every day at midnight, a new product was introduced, replacing the previous one, which was no longer available for purchase. This mechanism generated curiosity about what the next product would be, prompting users to visit the site on daily basis.

2.2.8 Core Drive 8: Loss & Avoidance

Loss & Avoidance is the *Core Drive* triggered when people want to avoid something negative from happening, like losing work, possession, or even opportunities.

Referring to what Daniel Kahneman said in his book "Thinking, Fast and Slow", people are on average twice as averse to the risk of losing something as they are seeking a possible gain [39]. In fact, we only take a risk if we think the potential gain is at least double what we could lose.

Yu-Kai Chou is keen to point out that in order to best implement this *Core Drive*, it is always necessary to present the user with an ultimate loss (more than 30% of what the user has spent in terms of time or resources, a big step backwards), but then only implement executable losses (much less than 30%, a much smaller step backwards).

Furthermore, the user must always know what to do to avoid the loss, otherwise this *Core Drive* backfires: the user enters denial mode and is motivated to avoid further engagement.

To give an example of how *Loss & Avoidance* is used, many social games employ it effectively. One of them, Farmville, is a simulation game that replicates various aspects of running a farm [40]. It encourages users to log into the game several times a day with the threat of losing the crops and livestock they have carefully taken care of.

2.2.9 Left Brain vs. Right Brain

Yu-Kai Chou groups the *Core Drives* according to the nature of the motivation they provide. Specifically, he classifies them into:

- **Left Brain** which includes *Core Drives* that focus on logic, analytical thought and ownership, and is the typical motivation derived from the desire to obtain something (*Core Drives 2, 4, 6*).
  They are characterized by an *Extrinsic motivation* which is derived from the goal, the purpose, or the reward. This type of motivation implies that the task itself is not necessarily interesting or appealing.
  The possible advantages of these *Core Drives* are the enhancement of focus on monotonous routine tasks and generation of initial interest for an activity.
  On the other hand, the disadvantages can be the reduction of creative and social capabilities, curiosity, out-of-the-box thinking, due to main focus on goal and rewards.

- **Right Brain** which includes *Core Drives* that focus on creativity, self-expression and social dynamics, and is involved in activities that are themselves rewarding (*Core Drives 3, 5, 7*).
  An *Intrinsic motivation* features these *Core Drives*, which is derived from the amusement and engagement derived from the task itself, without the need for any reward or compensation.
  The advantages of *Right Brain* can be the interest in the activity itself, due to creativity, social norm and curiosity, while the main possible disadvantage is lack of initial interest for the activity.
  To make an experience more intrinsic, Yu-Kai Chou argues that it should be modified to make it more social (for example, by allowing users to share their progress), to add unpredictability and randomness (being careful not to use *Core Drive 7* for too long and to make all the possible rewards appealing to the user), and to add meaningful choices (that reflect the user's style, preferences, and strategies).

2.2.10 White Hat vs. Black Hat

A different classification divides them into:

- **White Hat Gamification** includes *Core Drives* that make people feel powerful, fulfilled, satisfied, making them feel in control of their lives and actions (*Core Drives 1, 2, 3*). In general, these *Core Drives* are assigned a positive connotation.

The possible advantages are continuous increase of engagement, long-term retention and trust building, while the main disadvantage can be the absence of sense of urgency.

- **Black Hat Gamification** includes *Core Drives* that make people feel obsessed, anxious, addicted, making them feel like they don't control their lives and actions (*Core Drives 6, 7, 8*). In general, these *Core Drives* are assigned a negative connotation.
  The advantages can be the short-term revenue boost and low-cost initial increment of people engagement. On the other hand, the possible disadvantage is the limited long-term sustainability due to loss of control felt by people when they are objected of manipulative techniques.

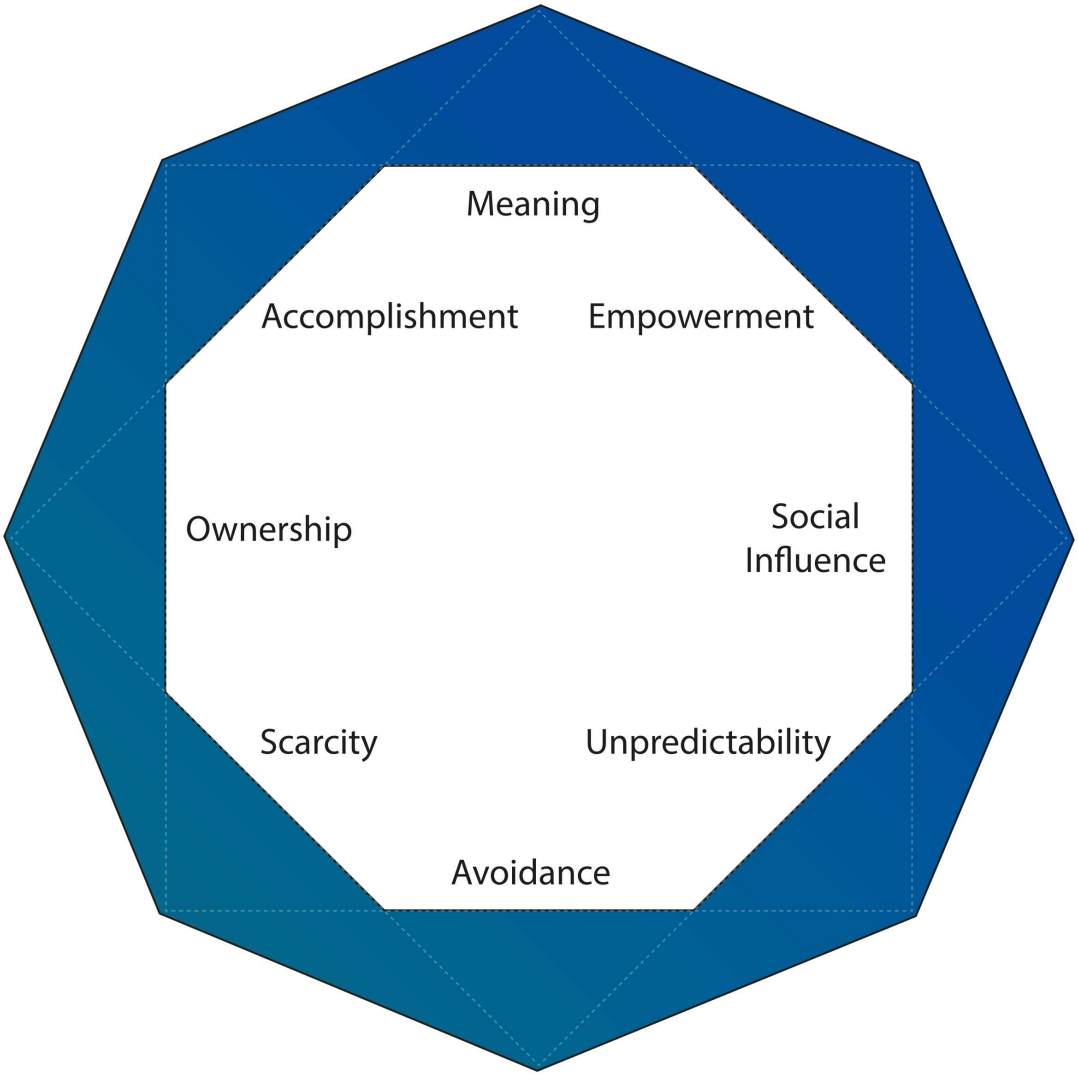

Figure 2.1: Octalysis Framework

Chapter 3

AwEE

AwEE is a browser extension that interactively displays the web navigation log, reporting in real-time the requests sent to known *Bad Hosts* from a blacklist, and showing related statistics. It also allows users to download the browsing log in different formats. To improve the user experience, a profile can be created, with the possibility of choosing a username and an avatar.

The name "AwEE" stands for "AwarE Extension", reflecting the tool's goal of raising user awareness about privacy concerns and web threats. The goal of this chapter is to provide a comprehensive overview of the software, both from a technological perspective and in terms of gamification choices.

3.1 Browser Choices

The proposed software is a browser extension, specifically developed as an extension for Firefox (also called Firefox add-on), a free and open-source web browser developed by the Mozilla Foundation that uses the Gecko rendering engine to display web pages.

The choice of Firefox was dictated by two main reasons: one purely practical and one more ethical. The first lies in the existence of MDN Web Docs, a comprehensive documentation repository and learning resource for Firefox web developers, maintained by the Mozilla Foundation [41], which explains in detail what extensions are, how they are structured, and how to build them.

The second reason, on the other hand, lies in the desire to sustain the open source community by developing a software compatible with its most popular and widely supported browser, in fact Firefox.

However, since according to recent statistics Firefox is currently used by a fairly small niche of people (around 2.5%, as is visible in Figure 3.1 [42]), the extension was also ported

to Chromium.

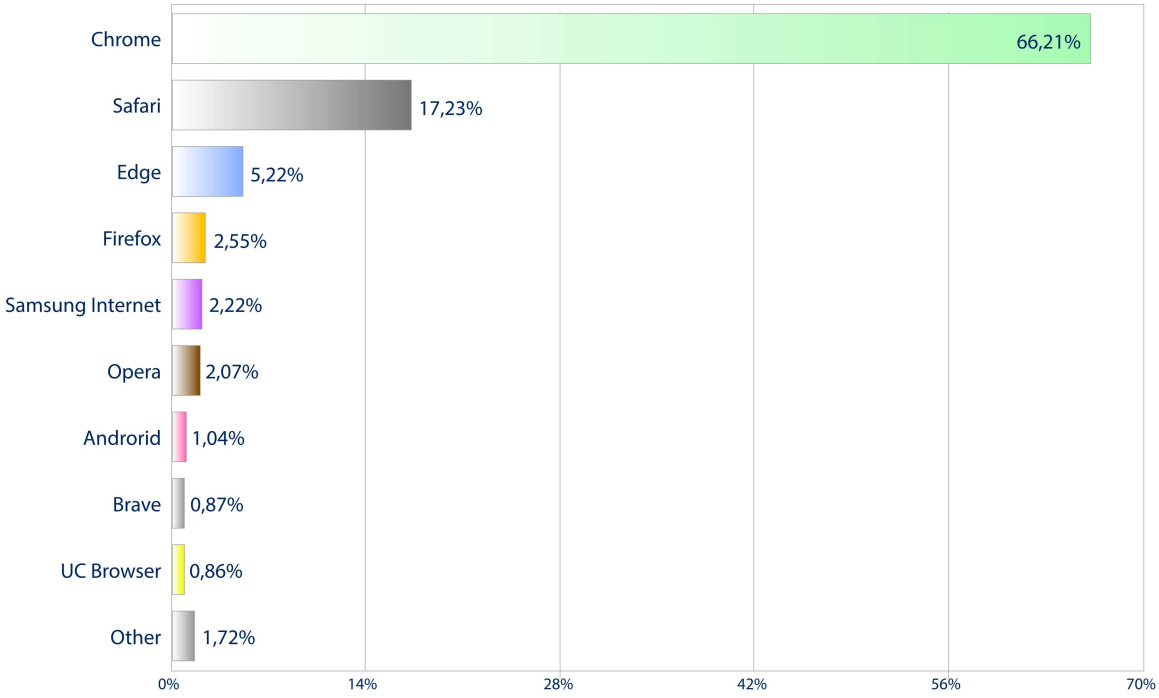

Figure 3.1: Data from StatCounter

Chromium is an open-source web browser project developed and maintained primarily by Google and based on the Blink rendering engine.

Most of the other popular browsers are based on this project, including examples such as Brave, Google Chrome, Microsoft Edge, and Opera.

Google has also defined WebExtensions APIs, a standardized set of JavaScript APIs that allow developers to build extensions able to interact with and modify browser functionality and web pages [43]. Firefox is, to a large extent, compatible with these APIs and therefore, in most cases, extensions written for Chromium-based browsers work in Firefox with only minor modifications and vice versa.

The steps required to perform the above-mentioned porting to Chromium were carried out by analyzing both the MDN documentation and the Chrome Extension Docs [44]. This ensures that the extension reaches almost all browsers used nowadays.

3.2 Architecture of the extension

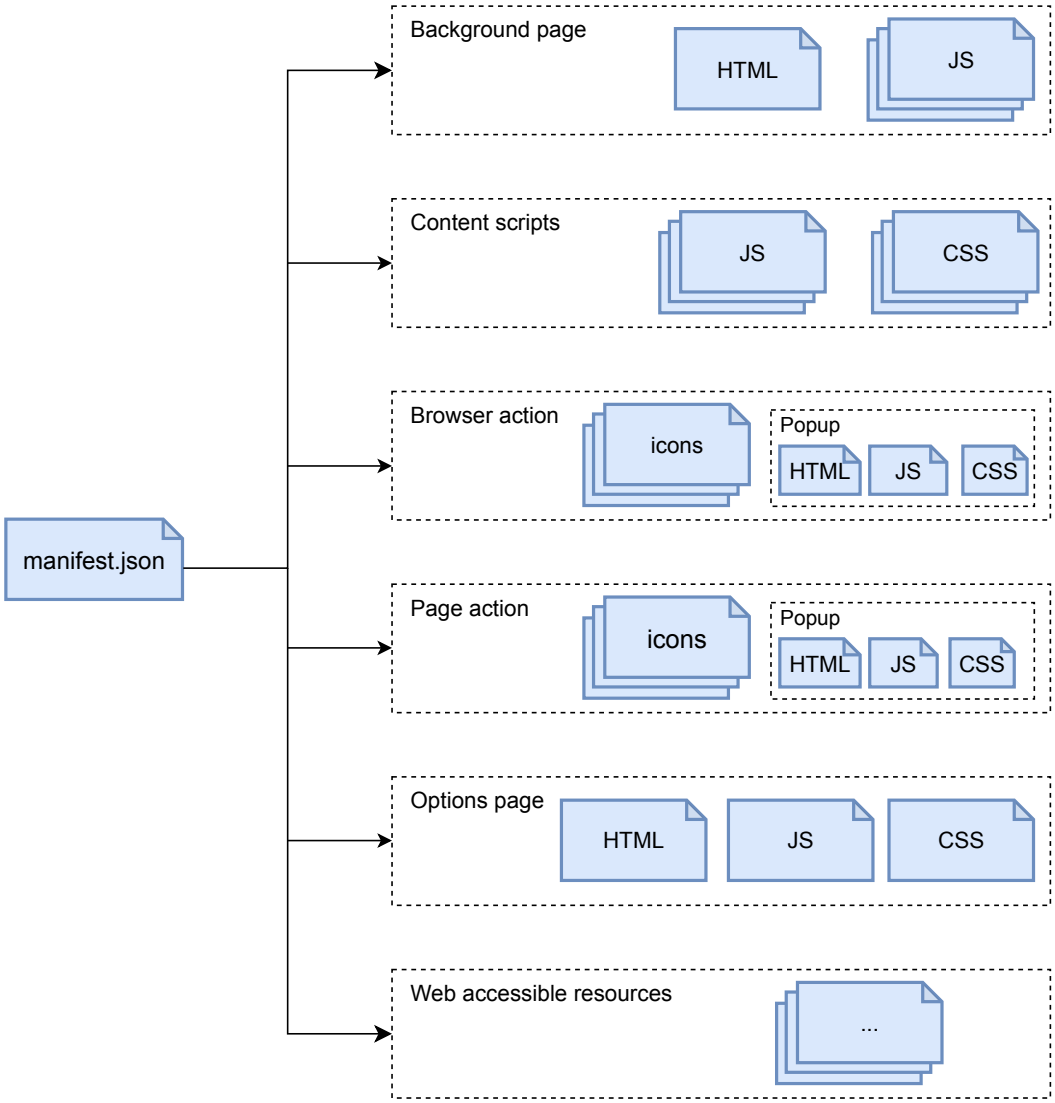

Figure 3.2: Anatomy of a generic Firefox extension

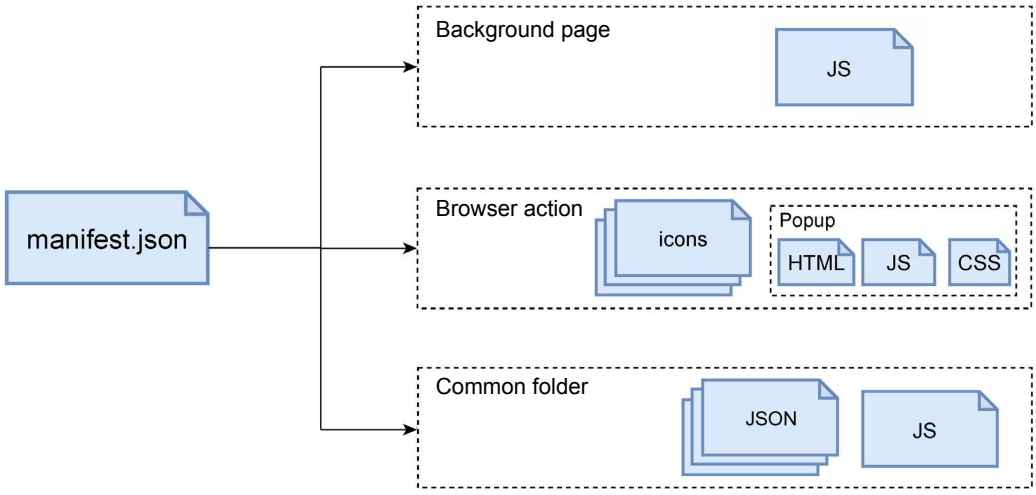

Figure 3.3: Anatomy of AwEE

AwEE is a browser extension and as such consists of a collection of files, packaged for distribution and installation (see Figure 3.2).

In this section, files that are part of the Firefox version of the project are briefly analyzed, using MDN as a reference.

As can be seen from Figure 3.3, the tool includes only a subset of the possible file types that can be included in a browser extension.

3.2.1 manifest.json

Code 3.1: Firefox version of AwEE: manifest.json

```
1 {
2     "manifest_version": 3,
3     "name": "AwEE",
4     "version": "1.0",
```

```
5
6     "description": "AwEE - The AwarE Extension",
7
8     "browser_specific_settings": {
9       "gecko": {
10        "id": "AwEE@polito.it",
11        "strict_min_version": "58.0"
12      }
13    },
14
15
16    "icons": {
17        "48": "icons/icon48.png",
18        "96": "icons/icon96.png"
19        },
20
21    "permissions": [
22        "downloads",
23        "webRequest",
24        "storage"
25    ],
26
27    "host_permissions": [
28        "<all_urls>"
29    ],
30
31    "web_accessible_resources": [
32        {
33          "resources": [],
34          "matches": ["<all_urls>"]
35        }
36    ],
37
38    "background": {
39        "scripts": ["background.js"],
40        "type" : "module"
41      },
42
43    "action": {
44        "default_area": "navbar",
45        "default_icon": "icons/icon48.png",
```

```
46         "default_title": "AwEE",
47         "default_popup": "popup/popup.html"
48         }
49 }
```

The *manifest.json* is the only file that must be present in every extension. It is crucial because it contains basic metadata such as extension's name, version, and manifest version. It provides pointers to other required files (in AwEE's case, background scripts and browser actions) specifying aspects of extension's functionality, and it defines required permissions. It is a JSON-formatted file, with one exception: it is allowed to contain comments introduced by "//".

In 3.1, the full content of the manifest.json file from the Firefox version of AwEE is reported.

It specifies that the extension complies with Manifest V3, the standard introduced by Google in 2018 for WebExtension APIs, aimed to modernize the extensions architecture and improve browser security and performance [45].

The *"permissions"* granted to the extension are the ones listed below:

- *"downloads"*, which allows user to start downloading files and is used to enable JSON/CSV export of the browsing log;
- *"webRequest"*, which provides the ability to intercept and act directly on HTTP requests made during navigation. It is used to intercept all web requests in order to analyze them and evaluate which ones are directed to *Bad Hosts*;
- *"storage"*, which offers the ability to store data persistently using the browser's storage API and is used to save browsing logs, user preferences, and gamification-related data.

The field *"host_permissions"* controls which websites the extension can access and interact with. In AwEE is set to *"all_urls"*, in order to be able to intercept network requests to any website.

"web_accessible_resources", on the other hand, controls which extension resources websites can access. Since in AwEE there is no need for sites to have access to any internal resources of the extension, this list of resources is set to be empty.

Finally, the *"background"* key defines the inclusion of the background script or Service Worker, while *"action"* specifies the files related to the toolbar button and its popup interface, which is how users interact with the extension.

3.2.2 Background page

The *Background page* (*background.js*), which in this case consists only in a Javascript file, is the script that enables to monitor and react to events in the browser. Background scripts can be persistent (loaded when the extension starts and unloaded when the extension is disabled or uninstalled) or non-persistent (loaded only when needed to respond to an event and unloaded when they become idle). In Manifest V2, non-persistent background scripts are recommended as they reduce resource cost of the extension, while, in Manifest V3, only non-persistent background scripts are supported.

AwEE complies with Manifest V3 and, consequently, its background script is a non-persistent one.

3.2.3 Popup

The *Browser Actions* in AwEE's case include the *icons* (for toolbar's extension button) and popup files. A popup is a dialog linked to a toolbar button or an address bar button. When the user clicks on the button, the popup appears. If the user clicks outside the popup, it closes. The popup is defined, similarly to a standard web page, using an HTML file (*popup.html*). It includes a CSS file (*popup.css*) describing the visual presentation and a JavaScript file (*popup.js*) which allows manipulating the DOM [46] and reacting to events via listeners. However, unlike a typical page, the JavaScript in the popup can access all the WebExtension APIs for which the extension has permissions. The popup's document is loaded each time it is displayed and unloaded whenever it is closed.

If *popup.html* is the page structure, defining the elements that make up the GUI, and *popup.css* deals with the visual appearance by adjusting the positioning and presentation of the elements defined in the HTML, *popup.js* handles all user interactions, displays real-time data, and coordinates with the background script.

3.2.4 Other resources

There are other local resources useful for AwEE operation.

The *common folder* groups inside some important files: *hosts.json*, a JSON file containing all the domains related to the *Bad Hosts*, *description-hosts.json*, a JSON file containing all the *Bad Hosts* descriptions, and the *utils.js*, a Javascript file containing definitions of functions used by other scripts.

The *img folder*, on the other hand, includes all the PNGs necessary for the user interface.

3.3 Bad Hosts

In AwEE, *Bad Hosts* are defined as companies or organizations that faced issues in managing user data, resulting in fines or ongoing investigations. As a result, *Bad Requests* turn out to be HTTP requests made to domains (we call them *Bad Domains*) that belong to a *Bad Host*.

In drafting the *hosts.json* file, the blacklist adopted within *Minos* was used as a starting point. This blacklist was compiled by volunteers from *MonitoraPA [47]*, a community that maintains an automated distributed observatory on the Italian Public Administration. They used *builtwith.com* [48], a web service that analyzes and identifies the technologies used by websites, to check the list of the most popular services in Italy referring to non-EEA countries.

Building the blacklist for AwEE, domains were grouped by their owning companies, allowing the extension to report the specific company associated with each *Bad Request*.

To justify the classification of these companies as *Bad Hosts*, the *description-hosts.json* file was created. In this file, each *Bad Host* is linked to a brief summary of the criticisms regarding the company's management of user privacy, along with a list of related references. Additionally, a compilation of significant news cases concerning fines or investigations related to the *Bad Host* were included.

Code 3.2: A snippet of *description-hosts.json* content

```
{
    "Google": {
      "description": "Google is an American multinational public corporation invested
          in Internet search, cloud computing, and advertising technologies.\n Its
          stated mission is \"to organize the world's information and make it
          universally accessible and useful\".\n This mission, and the means used to
          accomplish it, have raised concerns about, for example, the use of others'
          intellectual property and the ways Google collects data that may violate
          people's privacy.",
      "links": [
        "https://en.wikipedia.org/wiki/Criticism_of_Google",
        "https://en.wikipedia.org/wiki/Privacy_concerns_with_Google",
        "https://policies.google.com/privacy",
        "https://www.gnu.org/proprietary/malware-google.html"
      ],
      "newsCases": [
        "https://ec.europa.eu/commission/presscorner/detail/en/ip_19_1770",
        "https://apnews.com/article/google-incognito-mode-tracking-lawsuit-settlement-8
```

```
            b30c9397f678bc4c546ab84191f7a9d",
13          "https://www.politico.eu/article/google-hit-with-european-privacy-probe-over-
                its-artificial-intelligence-system/",
14          "https://www.theguardian.com/technology/2019/jan/21/google-fined-record-44m-by-
                french-data-protection-watchdog",
15          "https://www.bbc.com/news/technology-55259602"
16        ]
17      }
18  }
```

To explain concretely how the choice of *Bad Hosts* was justified, the case of Google is used by way of example. As can be seen in 3.2, The company is briefly described and then two lists are provided: a list of web references to provide more context regarding the company's description and concerns about the privacy of its users; and a list of links to news articles documenting specific cases that have resulted in fines or investigations for the company.

The latter, in Google case, includes:

- The €1.49 billion fine imposed by the European Commission in 2019 for violating EU antitrust rules by abusing its dominant market position with third-party websites, that prevented Google's competitors from placing their search ads on those websites [49];
- The already mentioned 2020 €5 billions fine for secretly having tracked users while using the incognito mode [50];
- The investigation by the Irish Data Protection Commission into Google's Pathways Language Model 2 (PaLM 2) to assess its compliance with EU data privacy laws [51];
- The €50 millions fine imposed by France's data protection authority on Google for violating the EU's General Data Protection Regulation (GDPR), since the company failed to provide users with clear and accessible information about data processing for personalized advertising, hindering informed consent [52];
- The £91 millions fine by French data privacy watchdog CNIL, because the company website did not ask visitors for their consent before saving advertising cookies on their computers [53].

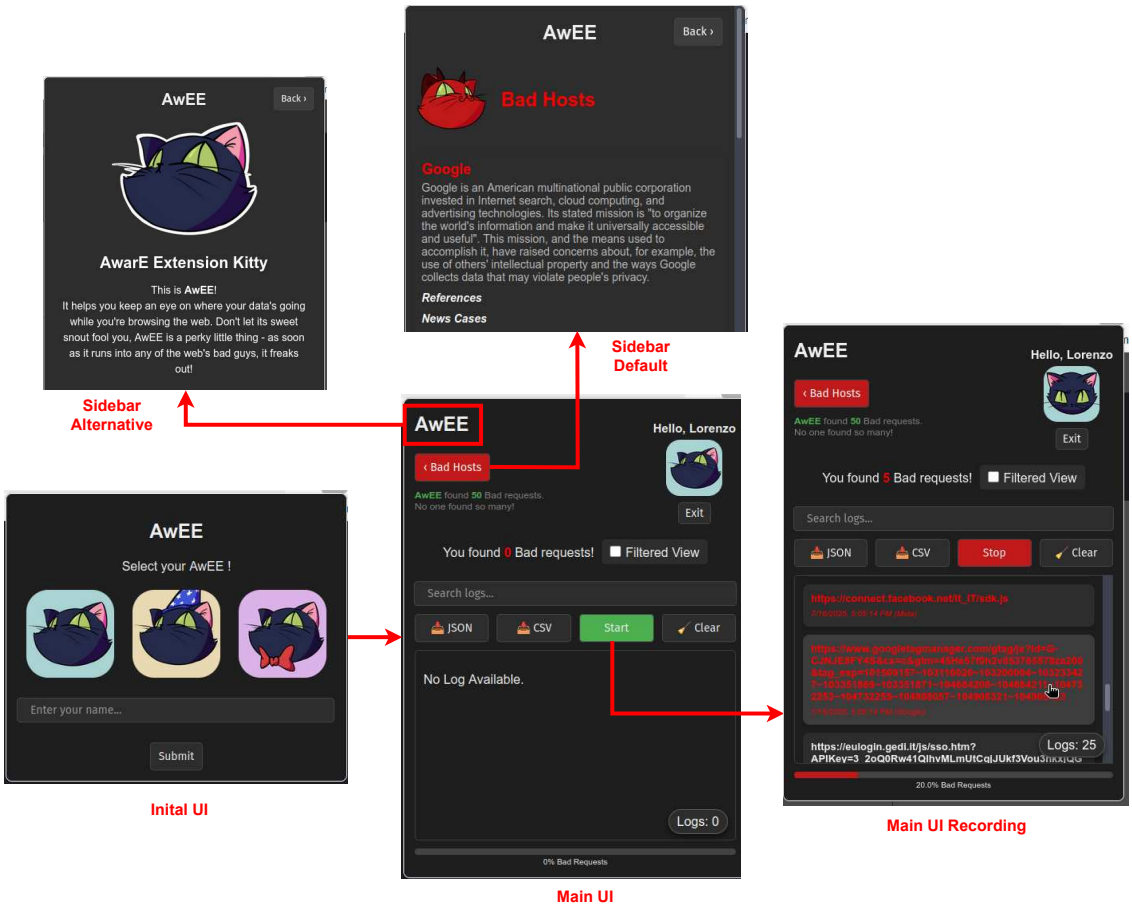

Figure 3.4: AwEE Interaction Workflow

3.4 GUI and Workflow

The goal of this section is to describe the GUI of AwEE, which is depicted in Figure 3.4, to guarantee an adequate understanding of the proposed software interface. Upon clicking the AwEE icon in the browser toolbar, a popup window appears for the user. The visual structure of this page is based on two main sections. When the users open the extension for the first time (or after a logout), they are presented with the *Initial UI*. The *Initial UI* allows the user to choose an avatar from a set of options and enter a username. After selecting an avatar and entering a username, the users click on the *Submit Button* which triggers the transition to the *Main UI*. The *Main UI* provides the main interface for interacting with the extension's functionalities.

The *Main UI* includes several key components.

The *User Profile* section at the top right displays the chosen avatar and username. The *Navigation Log Window* is the primary area where all HTTP requests made during browsing are listed. Users can filter these requests by using the *Search Bar*, which allows them to enter specific search terms, and the *Filter Option*, which enables filtering by *Bad Hosts*' domains listed in the *hosts.json* file.

Additionally, the interface includes several *Control Buttons*. The *Exit Button* logs the user out and clears its stored data, reverting to the *Initial UI*. The *Start/Stop Button* allows the users to begin or end the recording of their web navigation activity. When recording is activated, each navigation event (HTTP request) is displayed inside the *Navigation Log Window* along with timestamps. *Bad Requests* are highlighted in red and, next to the timestamp, the name of the *Bad Host* linked to the specific HTTP request is shown. Clicking one of these red entries triggers a special *Sidebar* opening, displaying detailed data from *description-hosts.json* about the specific *Bad Host* (Figure 3.5).

It is possible to download navigation logs in either JSON or CSV format using the *Download Buttons*, while the *Clear Button* resets the *Navigation Log Window*, clearing all displayed entries.

The collapsible *Sidebar*, initially hidden, can be opened by clicking on the red *Bad Hosts button* or on the *AwEE* title at the top left. The *Sidebar* provides two alternative contents based on the context. The *Default content*, triggered on *Bad Hosts button* pressure, shows detailed information about *Bad Hosts*, basically displaying details contained inside the *description-hosts.json* file. On the other hand, the *Alternative content* triggered on *AwEE* title pressure, shows additional insights related to AwEE's functionalities, and includes a list of recommended links to improve user's knowledge and awareness about privacy (*Privacy Guides* [54], *EFF* [55], *PrivacyTools* [56], *watchyourhack* [57], *Digital Defense* [58], *LeAlternative* [59], *PrivaSì* [60], *Cisti* [61]).

Throughout the interface, various *Counters* are also available. The *Log Counter* tracks the total number of HTTP requests and the *Bad Requests Counter* tracks how many of these requests were made to *Bad Hosts*. Meanwhile, *Best Score* and *Best Scorer* indicators highlight the highest number of *Bad Requests* logged by a single user and display that user's name.

Finally, on the bottom of the GUI, a *Progress Bar* visually represents the percentage of *Bad Requests* out of the total number of HTTP requests made.

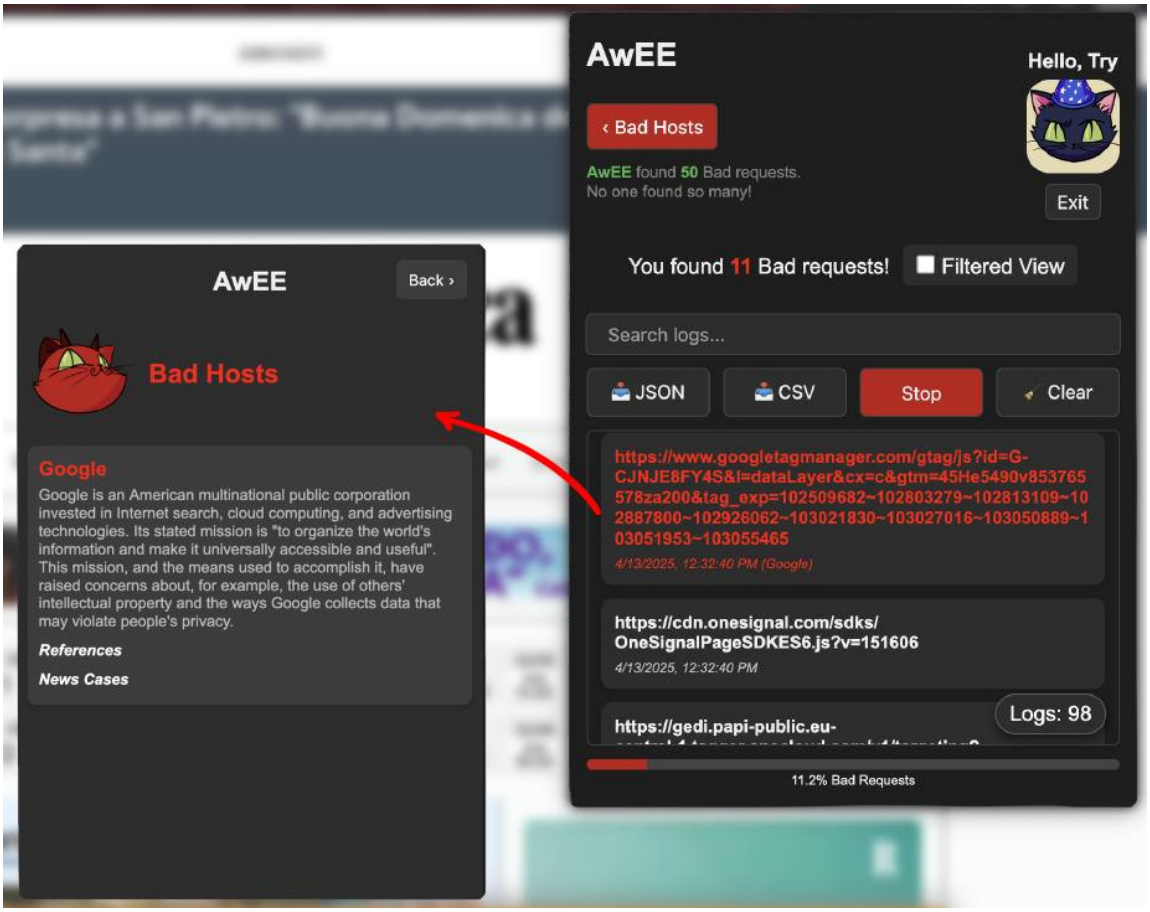

Figure 3.5: Check of a Bad Host during navigation

3.5 Gamification Techniques

In this subsection, the gamification-related additions to the extension are described, explaining how some techniques conceived by Yu-Kai Chou are exploited. Refer to chapter 2.2 for details involving the *Octalysis* framework's *Core Drives*.

3.5.1 Engaging Narrative

Engaging Narrative is a technique (triggering *Core Drive 1: Epic Meaning & Calling*) that uses storytelling to provide context and meaning to an experience. By weaving a narrative into a product or service — similar to how games set the stage with a plot or a quest

— the technique motivates users to become part of a larger purpose.

The power of this method was harnessed by introducing a dedicated page (the *Alternative Content* in the *Sidebar*) that tells about *"AwEE, the AwarE Extension Kitty"*.

This brief description of AwEE provides users with a compelling context that makes interacting with the tool more meaningful and this narrative element transforms the otherwise utilitarian experience into a more personalized and engaging journey.

3.5.2 Leaderboard

Leaderboard is a technique for motivating users by ranking them based on actions that lead to desired outcomes, and that triggers *Core Drive 2: Development & Accomplishment*. However, if the rankings are set too high or too broad, they can discourage new users by making the goal seem unattainable.

Effective *Leaderboards* create what's called *Urgent Optimism* by positioning users within reach of immediate improvement.

In AwEE, a *micro-leaderboard* was implemented, in the form of the record reporting the user who discovered the highest number of *Bad Requests* on the local machine where the extension is installed (see Figure 3.6). This allows users to compare their current performances only with their past sessions or with a small group of users from the same PC, instead of comparing against an overwhelming number of other people.

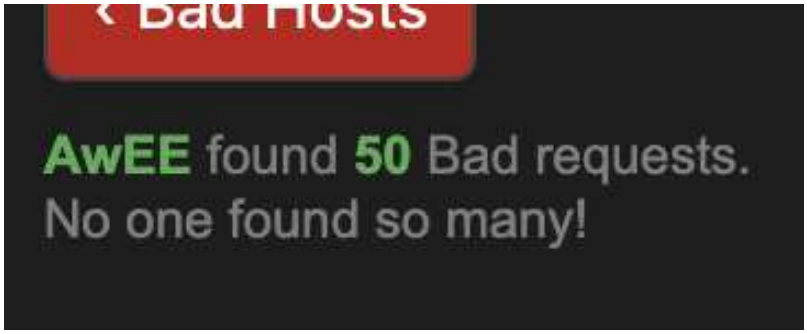

Figure 3.6: Micro-leaderboard record

3.5.3 Build-from-scratch

Build-from-scratch is a technique that triggers *Core Drive 4: Ownership & Possession*. It engages users by involving them in the creation process of a product or service rather than simply handing over a pre-made experience.

This technique was effecively applied by giving the possibility, before starting to use the extension's functionalities, to select an avatar from a set of three options, each featuring AwEE the kitty but with different accessories (see Figure 3.7). This initial decision empowers users with a feeling of control and personalization from the very start, but it prevents them from feeling overwhelmed because of too much choice.

(a) avatar 1 **(b)** avatar 2 **(c)** avatar 3

Figure 3.7: Avatar selection options

3.5.4 Dynamic Feedback

Dynamic Feedback is a technique that enhances user engagement by providing immediate, personalized responses as users perform desired actions. This method leverages real-time feedback to create a more interactive experience, encouraging users to continue keep using the interface. It triggers *Core Drive 3: Empowerment of Creativity & Feedback* (based on what they do, users see custom feedback) and *Core Drive 7: Unpredictability & Curiosity* (makes users want to find out what the next *Dynamic Feedback* will be). The technique was implemented by modifying users' selected avatar appearance in real-time: if users trigger a certain number of *Bad Requests* while they are navigating, the avatar's expression changes dynamically to reflect increasing levels of anger (see Figure 3.8). This visual feedback is linked to three thresholds of the number of *Bad Requests*.

Moreover, the extension's *Navigation log window* displays all HTTP requests as they occur, highlighting *Bad Requests* in red for clear visual emphasis and providing users with immediate responses to their actions.

(a) angry level 1 **(b)** angry level 2 **(c)** angry level 3

Figure 3.8: Avatar dynamic change

3.5.5 Monitor Attachment

Monitor Attachment is a method that encourages users to develop a sense of ownership by constantly overseeing the state of a system. When users regularly monitor progress such as by tracking numerical values or statuses, they naturally become more invested in its improvement. This ongoing attention not only makes the experience more engaging but also reinforces personal commitment, triggering *Core Drive 4*. This technique was implemented by giving users two critical counters in real time: one tracking all HTTP requests (Figure 3.9) and the other specifically counting *Bad Requests* (Figure 3.10). Additionally, a *Progress Bar* (Yu-Kai Chou defines it as a freestanding technique, triggering *Core Drive 2*) displays the percentage of *Bad Requests* out of the total HTTP Requests, offering clear and immediate insight into the system's current state (see Figure 3.11).

Figure 3.9: HTTP requests counter

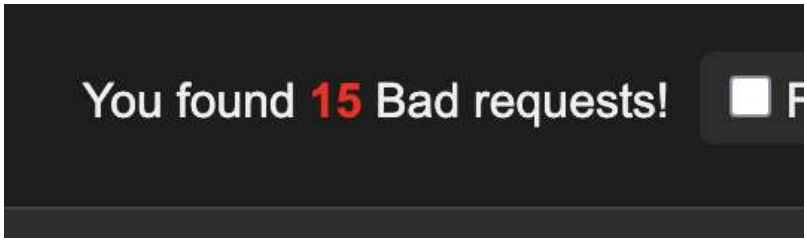

Figure 3.10: Bad Requests counter

Figure 3.11: Progress Bar

3.5.6 Octalysis Tool evaluation

In Figure 3.12 the output of the *Octalysis Tool* is reported, a project evaluation tool available on Yu-kai Chou's website that allows checking for compliance with the Octalysis Framework [62].

This visualization maps AwEE against each of the eight *Core Drives*, offering a clear outline of how the project scores under the *Octalysis Framework*'s criteria.

It shows that the focus was especially on *Left Brain* and on *White Hat Gamification*. Therefore, it follows that in addition to enhancing the above-mentioned areas, possible future improvements can be made particularly in the *Core Drives* of *Right Brain* and *Black Hat Gamification*.

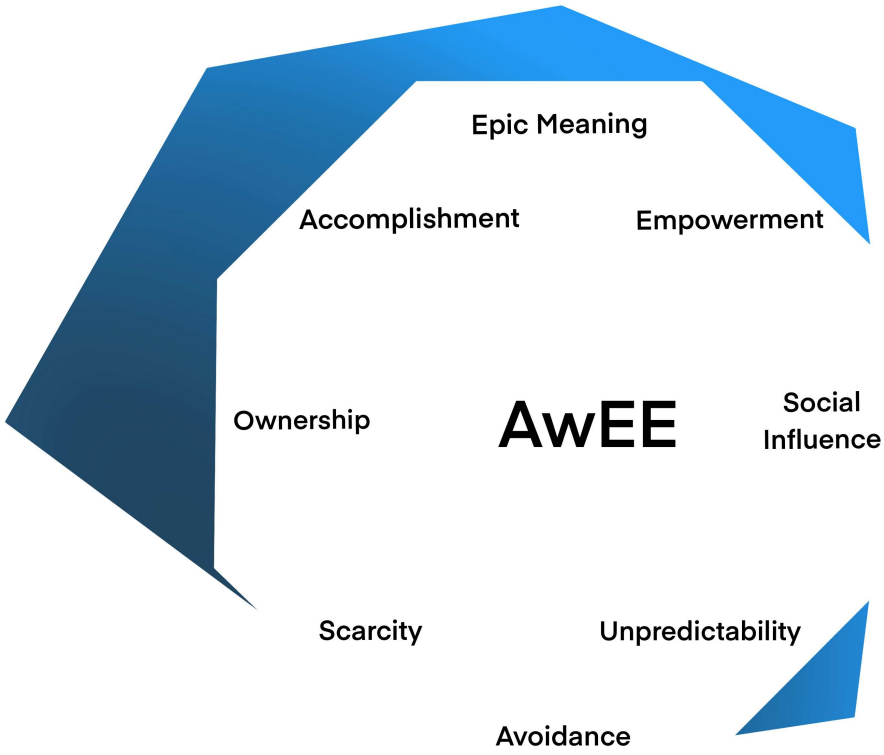

Figure 3.12: AwEE's Octalysis analysis

3.6 Behind the scenes

After understanding how AwEE works, how its graphical interface is structured, and what gamification mechanics characterize it, some technical details need to be clarified regarding how the above-mentioned features have been implemented.

3.6.1 Coordination between *background.js* and *popup.js*

To guarantee the correct functioning of the extension, certain specific information is stored persistently so that it is always available and can serve as communication between the background script and the popup interface.

To this end, the *storage* API is used, which allows extensions to store and retrieve data, as well as detect any changes to stored items [63]. Stored values can be strings, arrays, and objects, but only when their content can be represented as JSON files. When using *storage*, it is also necessary to specify its *properties*, which represent the different types of storage areas available. In the case of AwEE, only *storage.local* is referenced, which represents the local storage area, where items are local to the machine on which the extension is installed on. Unlike *storage.session*, where items are stored in memory for the duration of the browser session and are not persisted to disk, in *storage.local* objects remain in memory until the extension is uninstalled.

However, the use of locally stored shared variables alone does not guarantee proper coordination between *background.js* and *popup.js* to maintain real-time synchronization: they must agree on the methods and timing for managing access and modification to these resources.

To do this, the *runtime* module is used, which provides information about the extension and the environment it's running in, but also guarantees messaging APIs enabling to communicate between different parts of the extension itself [64].

Specifically, the function *runtime.sendMessage()* and listeners to the *runtime.onMessage* event are used to implement message exchange between *background.js* and *popup.js*.

The variables stored using *storage.local* are described below, explaining their usefulness and how access to their content is managed, also leveraging message exchange mechanism. This information is summarized in Table 3.1 and Table 3.2.

Recording state

Recording state is a boolean variable that keeps track of whether the user has initiated navigation registration using the appropriate button in the GUI.

popup.js is not authorized to modify it: it simply sends the *toggleRecording* message to *background.js* whenever the user clicks on the *Start/Stop* button.

Once the background script receives this message, it then takes care of updating it.

It is also responsible for initializing the variable to default *false* value.

Filter state

The *Filter state* boolean variable keeps track of whether the user has selected the navigation filter by domains related to Bad Hosts.

popup.js directly sets it on *True/False*, based on what user has selected in the related checkbox in the graphical interface. The popup uses it to know if the navigation log received from the background script needs to be filtered when being rendered in GUI (more information about this mechanism in 3.6.2).

background.js only initializes this *Filter state* to default *false* value.

Username and Avatar

Username contains the username entered by the current user, while *Avatar* memorizes the path to the picture of the avatar chosen by the current user.

While *background.js* initializes them to *none* and always resets them to that default value when it receives the *resetUserInfo* message from *popup.js*, the latter modifies them directly, based on what the user inserted in the *Initial UI*.

Best score and Best Scorer

Best score and *Best scorer* variables store respectively the record of the highest number of bad requests made and the username of the user who obtained it.

popup.js directly modifies their values when counters in the GUI overcome current record, while *background.js* is limited to initialize them (default values are *50* for *Best score* and *AwEE* for *Best scorer*).

Navigation Log

Navigation Log is an array containing the list of all HTTP requests recorded up to the current time while browsing.

The only one authorized to modify this array is the background script, which maintains it updated with user navigation (3.6.2 explains how). Every time *background.js* adds a new log entry, it tries to understand if popup is opened by sending it a **isPopupOpen** message and waiting for a *Yes* message. If it is, then the background script sends a *updateNavigationLog* message to *popup.js*, alerting it to check again the *Navigation Log* variable, otherwise it does nothing.

background.js also clears the log content and starts its download when requested by *popup.js* through respectively *clearLog* and *downloadJSON/downloadCSV* messages (download mechanism itself is described in 3.6.3).

Variable	Popup.js	Background.js
`Username & Avatar`	✓ User Profile Setup	✓ Init. & Reset
`Filter State`	✓ Checkbox Toggle	✓ Init.
`Best Score & Scorer`	✓ New record	✓ Init.
`Navigation Log`	× Read-Only	✓ Log Update
`Recording State`	× Read-Only	✓ Init. & Start/Stop Toggle

Table 3.1: *storage.local* write responsibilities between *background.js* and *popup.js*

Background.js → Popup.js	Popup.js → Background.js
`isPopupOpen`	`Yes` *(answer to* `isPopupOpen`*)*
`updateNavigationLog`	`toggleRecording`
	`clearLog`
	`resetUserInfo`
	`downloadJSON/downloadCSV`

Table 3.2: Message exchanged through *runtime* API

3.6.2 HTTP requests collection, processing and rendering

Code 3.3: Reduced example of navigation log array saved in *storage.local*

```
[
    {
        "url": "https://www.wikipedia.org/portal/wikipedia.org/assets/img/sprite-
            e49fbf32.svg",
        "page": "https://www.wikipedia.org/",
        "timestamp": "2025-09-30T23:13:15.996Z",
        "badhost": ""
    },
    {
        "url": "https://www.youtube.com/sw.js",
        "page": "https://www.youtube.com/",
        "timestamp": "2025-09-30T23:13:21.992Z",
        "badhost": "Google"
    },
    {
```

```
15        "url": "https://connect.facebook.net/it_IT/sdk.js?hash=800556167
              a0afd49f086e77423fee7e6",
16        "page": "https://www.repubblica.it/",
17        "timestamp": "2025-10-01T15:24:42.247Z",
18        "badhost": "Meta"
19    }
20  ]
```

When a user visits a URL, if the *Recording State* is set on *true*, the background script captures the request via *webRequest.onBeforeRequest*, the API for adding event listeners whenever an HTTP request is about to be made [65]. The script checks whether the URL matches any of the known *Bad Hosts* using the *hosts.json* file stored locally in the *common* folder. It then updates the *Navigation Log* with the URL, timestamp, and any associated *Bad Hosts* (refer to 3.3 for an example of *Navigation Log*).

When *popup.js* accesses the *Navigation Log* in read-only mode, before displaying it (appropriately signaling requests made to *Bad Hosts*), it checks the *Filter State* and any search key entered by the user in the search bar of the GUI. Only after doing this, it renders the log, filtered appropriately.

3.6.3 Export of navigation log

Code 3.4: Firefox version of AwEE: download mechanism in *background.js* (JSON only)

```
1  chrome.runtime.onMessage.addListener(async (message) => {
2      if (message.type === 'downloadJSON') {
3          const navigationLog = await loadNavigationLog();
4          const filteredLog = filterLog(navigationLog, message.searchTerm, message.
               filters);
5
6          const logData = JSON.stringify(filteredLog, null, 2);
7          const blob = new Blob([logData], { type: 'application/json' });
8          const url = URL.createObjectURL(blob);
9
10         const downloadId = await browser.downloads.download({
11         filename: 'navigation_log.json',
12         url: url,
13         });
14
15         URL.revokeObjectURL(url);
16
```

```
17          if (browser.runtime.lastError) {
18          console.error('Download failed:', browser.runtime.lastError);
19          } else {
20          console.log('Download JSON started with ID:', downloadId);
21          }
22      }
23  });
```

Finally, the background script is the main responsible for the navigation log export mechanism (3.4). Once it receives the *downloadJSON* or *downloadCSV* message from the popup, which also embeds the search key entered by the user in the search bar at the time of the download request, as well as the *Filter State*, it generates a temporary downloadable file [66] in the media type [67] consistent with the request received.

It places in it the current content of *Navigation Log* taken from *storage.local* and then uses the API function *downloads.download()* to download it [68].

3.6.4 Changes for Chromium

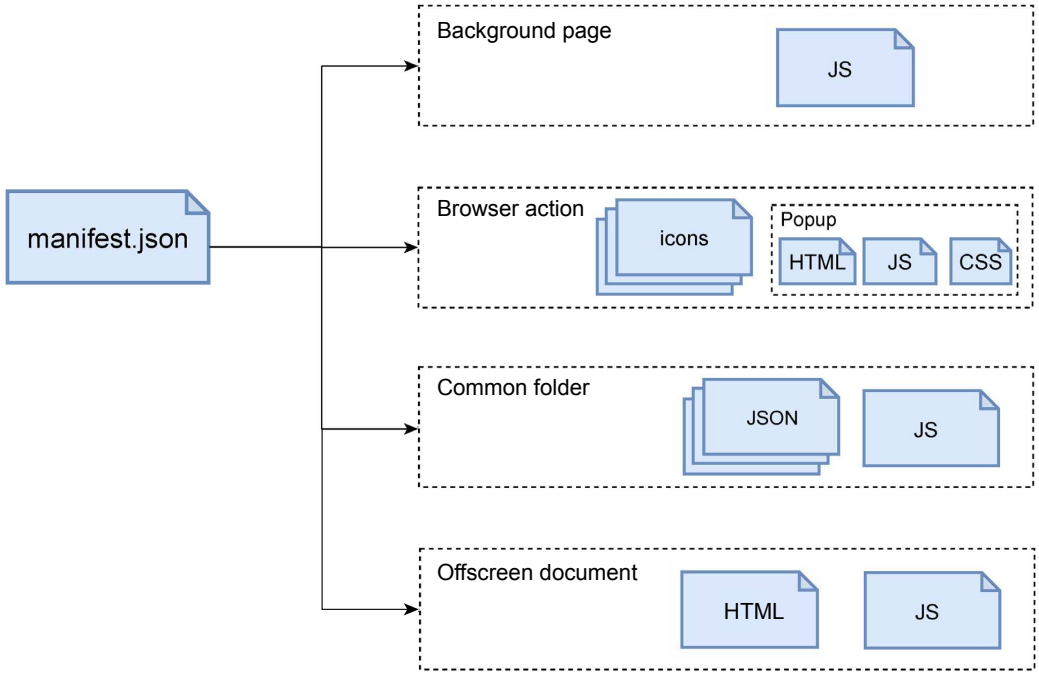

Figure 3.13: Anatomy of Chromium version of AwEE

Given the differences between Firefox and Chromium, in order to port AwEE to the latter, it was necessary to make some changes to the extension's code. This subsection aims to list and describe the main ones.

Manifest differences

Code 3.5: Chromium version of AwEE: a subsection of *manifest.json*

```
1 {
2     "manifest_version": 3,
3     "name": "AwEE",
4     "version": "1.0",
5
```

```
    "permissions": [
        "downloads",
        "webRequest",
        "offscreen",
        "storage"
    ],

    "web_accessible_resources": [
        {
          "resources": ["offscreen.html", "offscren.js"],
          "matches": ["<all_urls>"]
        }
    ],

    "background": {
        "service_worker": "background.js",
        "type" : "module"
      }
}
```

As already mentioned, the Firefox version of AwEE supports Manifest V3, making it compatible with most WebExtensionAPIs.

However, the implementation of Manifest V3 in the two browsers has some significant differences, due to concerns arising from the introduction of this standard.

As warned also by the Electronic Frontier Foundation, the changes are actually harmful for privacy and security, as Manifest V3 restricts powerful existing extension APIs (notably the blocking webRequest API) and pushes developers toward a limited *declarativeNetRequest* API, compromising many ad-blocking tools [69].

For this reason, Mozilla has chosen to be more conservative, supporting both *blockingWebRequest* and *declarativeNetRequest* in order to give developers more flexibility and to keep powerful privacy tools available to users [70].

In practice, the differences are not limited to this, as can be seen in 3.5. It shows part of the manifest.json file for the Chromium porting of AwEE, containing the fields that have been modified compared to the Firefox version.

While Firefox has extended compatibility with background scripts, allowing them to be used only in the non-persistent variant [71], Chromium instead uses service workers. They replace the extension's background page to ensure that background code remains outside the main thread. This allows extensions to run only when necessary, saving resources [72]. For this reason, in Chromium version of AwEE, the *background.js* file is

no longer treated as a background page, but as a Service worker (as highlighted in the *manifest.json* 3.5).

Furthermore, given that the export of the browsing log and its download are managed differently in the Chromium version, it was necessary to add *"offscreen"* to the *"permissions"* granted in the manifest, and to add the files related to the offscreen document (*offscreen.html* and *offscreen.js*) to the *"web_accessible_resources"*.

Handling downloads

Since in the Chromium version of AwEE *background.js* is used as a service worker, it does not have direct access to the popup's DOM, making it impossible to maintain the download mechanism directly within it.

The *offscreen* API allows the extension to use DOM APIs in a hidden document without interrupting the user experience by opening new windows or tabs [73].

The *chrome.runtime* API is the only API for extensions supported by offscreen documents (note that, in the context of Chromium development, the namespace used for extension API calls also changes: *browser* for Firefox, *chrome* for Chromium).

Therefore, it was opted to add an offscreen document, defined by textitoffscreen.html, whose body contains only a reference to the JavaScript script *offscreen.js*.

So, once the service worker receives a *downloadJSON/downloadCSV* message from the popup, all it does is sending a *download* message to the offscreen document that incorporates the format in which to download the content and the content itself from the navigation log.

offscreen.js takes then care of the actual download, in a manner entirely analogous to what happens in the *background.js* of the Firefox version of AwEE.

Below the code related to what has been described is viewable (3.6, 3.7, 3.8).

Code 3.6: Chromium version of AwEE: download mechanism in *background.js* (JSON only)

```
1
2 async function createOffscreenDocument() {
3     const offscreen = await chrome.offscreen.createDocument({
4       url: chrome.runtime.getURL('offscreen.html'),
5       reasons: ['BLOBS'],
6       justification: 'Download file',
7     });
8     return offscreen;
9 }
10
11 function blobToBase64(blob) {
```

```
return new Promise((resolve, reject) => {
    const reader = new FileReader();
    reader.onloadend = () => {
    resolve(reader.result.split(',')[1]);
    };
    reader.onerror = reject;
    reader.readAsDataURL(blob);
    });
}

chrome.runtime.onMessage.addListener(async (message) => {
  if (message.type === 'downloadJSON') {
    const navigationLog = await loadNavigationLog();
    const filteredLog = filterLog(navigationLog, message.searchTerm, message.filters);
    const logData = JSON.stringify(filteredLog, null, 2);
    const blob = new Blob([logData], { type: 'application/json' });

    const base64Blob = await blobToBase64(blob);

    chrome.offscreen.closeDocument();
    const offscreen = await createOffscreenDocument();

    chrome.runtime.sendMessage({
      type: 'download',
      data: base64Blob,
      format: 'json',
    });

    console.log('Download JSON started');
  }
});
```

Code 3.7: Chromium version of AwEE: *offscreen.html*

```
<!DOCTYPE html>
<html>
<head><title>Offscreen Document</title></head>
<body>
    <script src="offscreen.js"></script>
</body>
```

```
</html>
```

Code 3.8: Chromium version of AwEE: *offscreen.js*

```
chrome.runtime.onMessage.addListener((message, sender, sendResponse) => {
    if (message.type === 'download') {
      const base64String = message.data;
      const format = message.format;
      const byteCharacters = atob(base64String);
      const byteArrays = [];
      for (let offset = 0; offset < byteCharacters.length; offset += 512) {
        const slice = byteCharacters.slice(offset, offset + 512);
        const byteNumbers = new Array(slice.length);
        for (let i = 0; i < slice.length; i++) {
          byteNumbers[i] = slice.charCodeAt(i);
        }
        byteArrays.push(new Uint8Array(byteNumbers));
      }
      const blob = new Blob(byteArrays, { type: 'application/json' });
      const url = URL.createObjectURL(blob);
      const a = document.createElement('a');
      a.href = url;

      if (format === 'json') {
        a.download = 'navigation_log.json';
      } else if (format === 'csv') {
        a.download = 'navigation_log.csv';
      }

      a.click();
      URL.revokeObjectURL(url);
    }
  });
```

Throttling of GUI update exclusion

In the Firefox version of AwEE, it has been empirically found that when many HTTP requests are forwarded simultaneously to be rendered in the popup, the UI becomes unresponsive due to small graphical lags.

In order to mitigate this situation, the frequency with which the popup checks the *Navigation Log* to update its rendering was limited, introducing a minimum time that must elapse between updates (*THROTTLE_TIME*, currently set to 1 second. See 3.9).

Since the same issues were not found in the graphical interface of the Chromium variant, *THROTTLE_TIME* was excluded from the code of this version.

Code 3.9: Firefox version of AwEE: Throttling of GUI update (*popup.js*)

```
let lastCalled = 0;
const THROTTLE_TIME = 1000;
async function updateNavigationLog(searchTerm = '') {
  const now = Date.now();

  // Check if the function was called too recently
  if (now - lastCalled < THROTTLE_TIME) {
    console.log("Aborted: updateNavigationLog was called too recently.");
    return;
  }

  lastCalled = now;
  // ... rest of the function
}
```

Chapter 4

Comparison with existing tools

There are several browser extensions available online that inspired AwEE, especially the ones allowing content blocking or providing information about user data sent to the servers of visited pages. The purpose of this chapter is to list and describe the most important ones, providing insights into how they eventually influenced the development of AwEE. The summary table 4.1 briefly recapitulates these contents.

Extension	Blacklist	Tracking Detection	Fingerprinting Detection	Gamification
NoScript	✓			
Privacy Badger		✓		
uBlock Origin	✓			
FPMON			✓	
AwEE	✓			✓

Table 4.1: Extensions Characteristics Comparison Table

4.1 NoScript

As already mentioned in 2.1, *NoScript* is a browser extension that allows JavaScript and other potentially harmful content to be executed only on trusted websites of user choice [74]. For every visited page, NoScript analyzes the DOM to identify active content such as scripts and multimedia objects. It detects the visited domains and compares them with three different sets of domains: a whitelist containing trusted domains whose scripts can be loaded, a blacklist containing domains considered dangerous, and a set of predefined rules. The lists are stored in the browser's local storage (JavaScript API provided by web

browsers to store data that persists after the browser is closed) and can be modified by the user through the GUI. Additionally, the same GUI also allows blocking or authorizing other specific types of content within the pages, such as media. In the implementation of AwEE, it served as a reference for how to incorporate a blacklist within a browser extension.

4.2 Privacy Badger

Made by the Electronic Frontier Foundation (EFF), *Privacy Badger* is a browser extension that blocks third-party trackers during the navigation [75]. It stands out from other blockers in two significant ways. First, it doesn't block ads unless the extension recognizes that they happen to be tracking the user. Second, it does not rely on a human-curated list of domains or URLs to block. Instead, it uses an algorithmic approach to define if a domain is tracking the user or not. The extension keeps track of the third-party domains (domains different from the one explicitly visited by the user) that embed images, scripts and advertising in the pages visited by the user. Privacy Badger looks for tracking techniques like uniquely identifying cookies, local storage cookies and canvas fingerprinting. If it observes the same third-party host tracking on three separate sites, Privacy Badger will automatically disallow content from that third-party tracker. This means that what is considered a tracker is determined by the domain's actions, not by human judgments.

4.3 uBlock Origin

uBlock Origin is a content blocking extension that allows to block ads, trackers, and fingerprinting scripts. It uses by default publicly available static blacklists permitting the user to easily choose which of them to enable or disable [76]. It is available for Firefox and Chromium, despite the fact that, due to Google's decision to completely deprecate extensions with Manifest V2 (such as precisely uBlock Origin), it will soon be unsupported by Chromium and already appears flagged by currently available versions of the browser as "soon no longer supported" [77]. It features two different modes: a Basic Mode, consisting of a simple popup interface for plug-and-play and default configuration installations, and an Advanced Mode, which instead includes an interactive, point-and-click interface that allows users to adjust specific settings for individual websites [78].

From a visual point of view, this second mode introduces a lateral sidebar within the popup interface, which displays additional content to the user, for interactive per-site settings.

This specific feature was a major inspiration for part of the AwEE's popup interface, in which the use of a sidebar, employed for multiple purposes depending on the context, turns out to be critical.

4.4 FPMON

FPMON is a browser extension (already mentioned in 2.1) able to measure and rate fingerprinting activity on any website in real-time [20]. The authors of this software first classified the Javascript functions typically used to fingerprint a device. Then, they implemented a mechanism that intercepts and records these functions without altering the default runtime behavior, by modifying the Javascript runtime environment with code injections. In the end, they designed two ways to present results to the user. First, through the browser extension icon, that represents a human fingerprint whose color changes according to the level of fingerprint activity detected. Second, through a detailed view of the overall analysis visible as a popup after pressing the icon. In addition, the top 3 script files that enabled most fingerprinting features are shown.

As can easily be imagined, this tool inspired AwEE to use techniques such as *dynamic feedback* and a *micro-leaderboard*.

Chapter 5

Evaluation Experiment

In this chapter, a detailed overview of the evaluation experiment conducted to assess AwEE's effectiveness and user engagement is presented. The evaluation took place during an Educational Guidance day organized at Politecnico Di Torino, Italy. The main purpose of this event was to offer advice to students in their last year of high school and to students completing their bachelor's degrees (on separate days). The focus was on university programs, career opportunities, and insights into how the Politecnico di Torino operates [79] [80].

Despite the event not being designed specifically for the evaluation of AwEE, it was seen as an ideal opportunity to gather voluntary feedback from the attendees. With this in mind, a brief questionnaire was submitted to gather insights from participants who chose to answer. A total of 120 participants took part in the evaluation, 61 of them were high school students and 59 were bachelor students.

In Figure 5.1 an exact copy of this questionnaire is displayed.

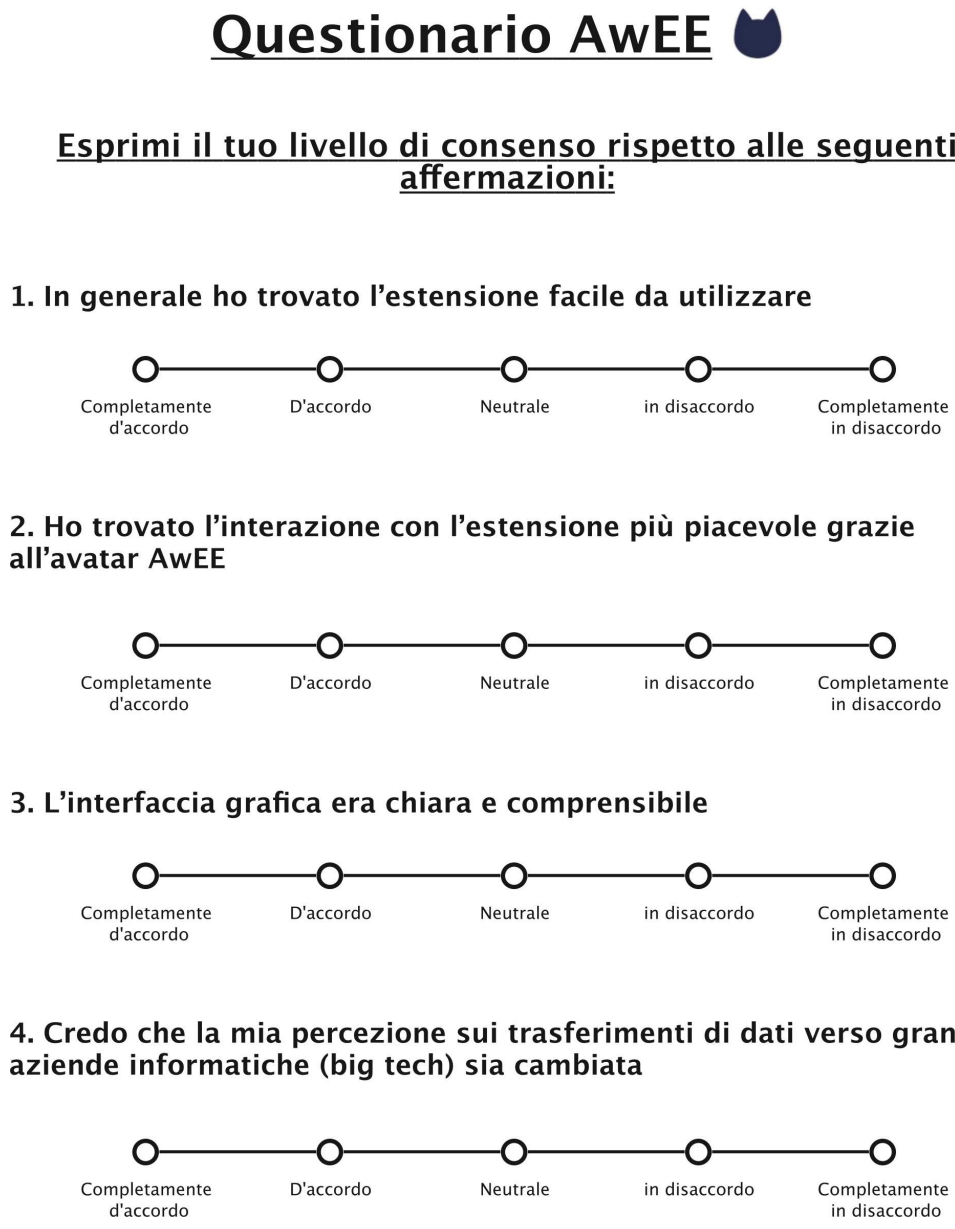

Figure 5.1: A copy (in Italian) of the questionnaire submitted to participants

5.1 Evaluation Method

Participants were introduced to AwEE and asked to use it during their web browsing. After interacting with the extension for a short period, each participant completed a questionnaire to assess their experience. The questionnaire consisted of 4 statements and the participant had to assign their level of agreement with each of them. Statements were designed to measure user engagement, ease of use, and the impact of the extension on their perceptions of data privacy.

The statements were:

Q1: *I found the extension easy to use.*

Q2: *I found the interaction with the extension more pleasant thanks to AwEE avatar.*

Q3: *The Graphical User Interface was clear and understandable.*

Q4: *I believe that my perceptions on data transfers to big tech companies have changed.*

Statements Q1, Q2 and Q3 were obtained by adapting the System Usability Scale [81] to our context, while Q4 is a domain-specific one.

For each of the four statements, there were 5 possible answers following a classic Likert scale structure [82], ranging from *Completely Disagree* to *Completely Agree*.

The questionnaire was deliberately kept short and concise to avoid overwhelming participants or distracting attention from the main objectives of the event, namely to provide guidance to high school and undergraduate students. Since the event organizers kindly allowed the experiment to take place, every effort was made to ensure that it coexisted with the aforementioned objectives of the *Polito Open Days*.

5.2 Results

Figure 5.2 and table 5.1 show the results of the questionnaire. Answers from high school students are labeled with **HS**, while answers from bachelor's degree students were labeled with **BS**.

These data obtained as a result of the experiment provide valuable insights into the effectiveness of AwEE in achieving its goals of improving privacy awareness and engaging users through gamification. As is clear from looking at Figure 5.2, and Table 5.1 above, more than 69% (41 out of 59) of bachelor students and more than 75% (46 out of 61) of high school students answered *Agree* or *Completely Agree* to the **Q4**. This means that the majority of participants believe that the use of AwEE changed their awareness

of data transfers to big tech companies. So, it shows that the tool can work as a first step towards improving privacy awareness among users.

Furthermore, the majority of both bachelor's and high school students agreed that the interface was clear and easy to understand, the extension was enjoyable to use, and its overall usability was high (with regard to **Q1**, **Q2** and **Q3**). This suggests that AwEE's design is fairly intuitive and accessible, while its functionalities are quite self-explanatory and pleasant to use.

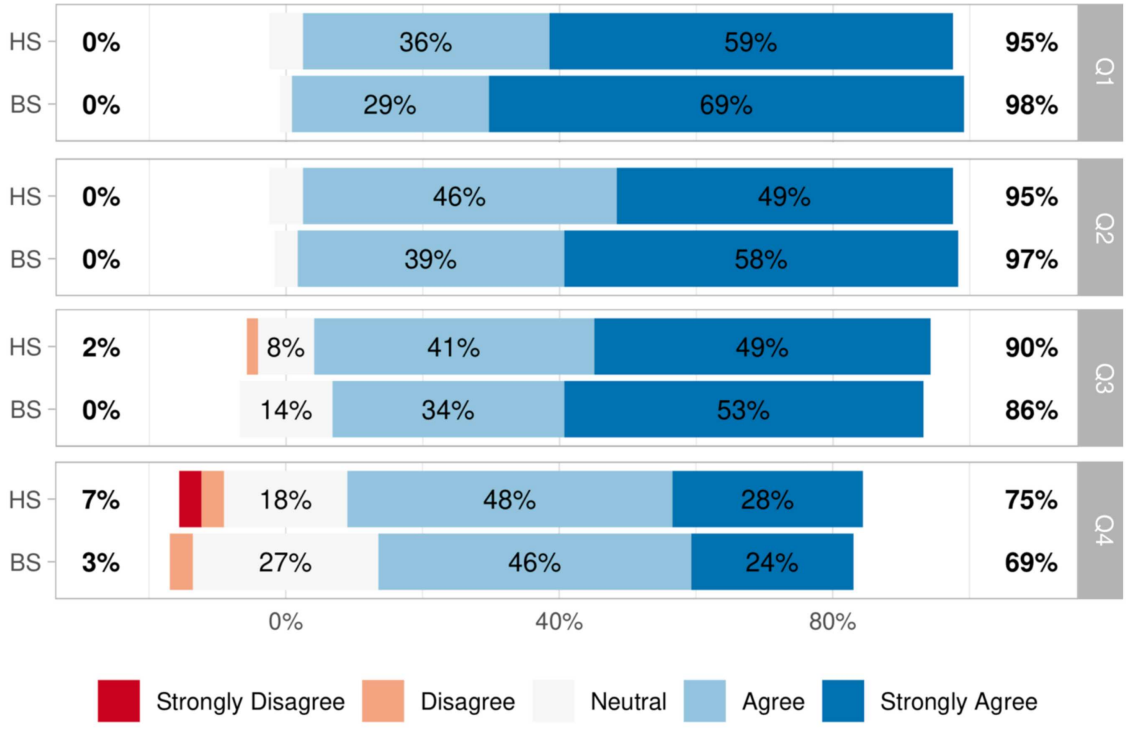

Figure 5.2: Results of the questionnaire

#Q	Students Category	Fully Agree	Agree	Neutral	Disagree	Fully Disagree
Q1	HS	36	22	3	0	0
	BS	41	17	1	0	0
Q2	HS	30	28	3	0	0
	BS	34	23	2	0	0
Q3	HS	30	25	5	1	0
	BS	31	20	8	0	0
Q4	HS	17	29	11	2	2
	BS	14	27	16	2	0

Table 5.1: Results of the questionnaire

In addition to the questionnaire, participants were encouraged to provide free-form feedback on their experience with AwEE. Here are the key points from their comments:

- A couple of students mentioned that the bright red text used inside the GUI (for example to highlight Bad Requests or their counter) made it harder for them to read the information in the log. In their opinion, it generally worsens the readability of the interface.
- Several participants were surprised that companies such as Apple, TikTok, Temu, OpenAI, Xiaomi, or Huawei were not considered *Bad Hosts*. This suggests that a potential future improvement could involve expanding the list of *Bad Hosts* evaluating the inclusion of these companies and others.
- Some users suggested incorporating a log of JavaScript execution during browsing, similar to what FPMON does.
- A couple of students mentioned that seeing the full HTTP requests in the log, especially when it's long and complicated, might confuse users who don't know what an HTTP request is. Instead of displaying the entire request, they suggested that the extension could just indicate the target host of the request.

Chapter 6

Threats to validity and possible solutions

Although AwEE shows promise as a privacy awareness tool, several factors limit the strength and generalizability of its results. The aim of this chapter is to examine the weaknesses of the project with a view to improving it and marking the way for possible future research.

6.1 Blacklist limits

Firstly, the way *Bad Hosts* are identified should be considered as a possible area of weakness. Maintaining a list that includes all the companies subject to criticism about their data management involves a lot of effort. Continuously researching every new privacy incident, fine, or legal ruling is a labor-intensive process. It also runs the risk of being highly biased by what the contributors to this list think and what their priorities are.

These problems are inherent to blacklists in general, since compiling a list of malicious domains, even using different techniques such as manual reports, web crawlers, and heuristic analysis of websites, remains a static approach.

Inevitably, this means that there will always be malicious sites that are overlooked, either because they are too new, because they have been incorrectly assessed due to a classification error, or because they weren't evaluated at all.

A possible alternative approach could be to develop an automatic method for classifying URLs, not unlike that proposed by *Ma et al.* [83].

Through a careful, less empirical redefinition of the concept of *Bad Host*, and the simultaneous definition of significant features on the basis of which to evaluate the various

URLs, machine learning models could in fact be used to classify websites, distinguishing between "malicious" and "non-malicious" ones.

On the contrary, if a more conservative approach is desired, it may be decided to rely on the use of a blacklist, but to compile it starting from an already existing one created with the explicit purpose of identifying malicious domains. Indeed, AwEE has currently adapted the blacklist used by Minos, whose original purpose was to identify requests directed to countries outside the EEA. However, lists such as *EasyList* [84], *Peter Lowe's Blocklist* [85], and *Online Malicious URL Blocklist* [86] (also used by *uBlock Origin*, among others) could be exploited. Alternatively, resources such as DuckDuckGo Tracker Radar, made available under an open source license by DuckDuckGo, could be used [87]. This is a continuously updated and constantly tested dataset that collects all the main existing trackers and descriptions of their tracking behavior.

6.2 Evaluation Limits

The preliminary feedback received from users also has its limitations. In the context in which the questionnaire was offered, participants were aware that they were helping to test a new tool and may have been inclined to offer positive comments, showing confirmation bias. To avoid overwhelming the students, and not to shift the focus too much from the main objectives of the event, the survey was also kept extremely short. Thus, while a useful snapshot of first impressions were obtained, participants' opinions couldn't be explored more deeply.

Therefore, it might be worth considering conducting much more in-depth assessments on larger groups of volunteers, who would be given the opportunity to interact with AwEE in a less superficial way, potentially analyzing its use in users' everyday lives and the effects it could have on them.

6.3 Gamification limits

Finally, regarding AwEE's gamification features, limitations of what was implemented must be highlighted.

Certainly the "micro-leaderboard" offers a too limited local leaderboard to stimulate meaningful competition. Moreover, the short narrative of *"AwEE, the Aware Extension Kitty"* is located in the *Sidebar* and exploring it is not essential or particularly stimulating concerning the usage of the extension. To improve this context, for example, a background storyline could be created, which should explain the importance of privacy

with a character or group of characters representing privacy-conscious entities. It would trigger *Core Drive 1: Epic Meaning & Calling* in a deeper way than is currently the case.

As also reported in Section 3.5, the *Right Brain* and *Black Hat* motivators were neglected, while, according to Yu-Kai Chou, they can greatly increase long-term engagement. Without improving this kind of motivators, we may have difficulty sustaining user interest beyond the initial novelty. For instance, introducing the possibility to unlock new customizable avatars as users reach milestones or accomplish certain tasks would add a layer of personalization and curiosity, tying into *Core Drive 7: Unpredictability & Curiosity*, in addition to *Core Drive 4: Ownership & Possession*.

A point assignment system could be implemented, where users earn *privacy points* for taking actions that contribute to their privacy awareness, such as identifying *Bad Hosts* or learning about privacy risks, and lose them for acting risky, like interacting with known *Bad Hosts*. Developing a points' economy would trigger again *Core Drive 7* and *Core Drive 4* as well as *Core Drive 2: Development & Accomplishment*. Another valuable feature could be the option for users to create personalized blacklists and whitelists of websites or companies they trust, or don't trust (*Core Drive 3: Empowerment of Creativity & Feedback*), which then could be shared with others, encouraging social influence and relatedness (*Core Drive 5: Social Influence & Relatedness*).

Chapter 7

Conclusions and Future improvements

This thesis contributes a novel approach to enhancing privacy awareness by presenting a new browser extension with integrated gamification mechanics, called AwEE.

The extension was designed, implemented, and preliminarily evaluated, assessing both its usability and its ability to shift users' perceptions about data transfers to big tech companies. The tool logs HTTP requests made by users during their navigation, highlighting which of them refer to *Bad Hosts* taken from a blacklist. Users are motivated thanks to several gamification elements such as interactive statistics, dynamic avatar feedback, a micro-leaderboard, and a simple customization of user profile.

As a first step for this project, already existing research and work focusing on privacy and gamification were listed and described. Next, the extension itself was depicted: its architecture, how it works, its interface, implementation details, and the technical choices made. The gamification principles used within the tool were also discussed, explaining the reasons for their use. After comparing AwEE with existing privacy tools, a preliminary evaluation experiment of AwEE was explained, commenting on the results obtained. The outcome of such assessment highlights both extension's usability and ability to shift users' perceptions about data transfers to big tech companies, demonstrating the potential of combining privacy tools with gamification.

However, there are several areas where AwEE could be further improved to enhance its functionalities and its capability to commit users.

AwEE currently relies on a static blacklist to identify *Bad Hosts*. A potential improvement would be to explore alternative methods to automatize the recognition of *Bad Hosts* and eventually implement them. Another possibility is the introduction of a server-based backend. Although keeping everything local on the user's device guarantees privacy,

a server-based system could enable better user account customization, more extensive leaderboards, sharing and communication features between users. However, this approach would require to implement a GDPR-compliant privacy policy to ensure the safe handling of user data, and managing the server infrastructure would add complexity and potential costs.

Consideration could be given to conducting new experiments to evaluate the validity of the project in greater depth, which could potentially provide new ideas for improvements and further research.

Additional gamification mechanics can also be implemented in the tool. For example, introducing a background story involving characters representing privacy-conscious entities could transform AwEE into a fully immersive experience. Consequently, introducing the ability to unlock customizable avatars when users reach some milestone would increase their engagement by intriguing them and by giving them powers of personalization. A points-based system could also be implemented that rewards users with privacy points for actions that improve privacy and deducts points for risky behavior. Furthermore, a feature could be offered that allows users to create and, optionally, share personalized blacklists and whitelists of websites or companies they trust or distrust.

Finally, some aspects of AwEE's GUI would be modified to improve its readability, following student advices given during the preliminary evaluation experiment.

By focusing on these areas, AwEE could continue to evolve as a powerful tool for promoting privacy awareness while keeping users motivated and engaged in safeguarding their personal data.

Bibliography

[1] Shoshana Zuboff. *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power*. 1st edition (2018). ISBN 1610395697.

[2] Nick Couldry and Ulises Ali Mejias. *The costs of connection: how data is colonizing human life and appropriating it for capitalism*. Culture and economic life. Stanford University Press, Stanford, [California] (2019). ISBN 978-1-5036-0366-0 978-1-5036-0974-7.

[3] The Guardian. https://www.theguardian.com/world/2013/jun/09/edward-snowden-nsa-whistleblower-surveillance, *Edward Snowden: the whistleblower behind the NSA surveillance revelations* (2013).

[4] The Guardian. https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election, *Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach* (2018).

[5] The Guardian. https://www.theguardian.com/technology/2024/apr/01/google-destroying-browsing-data-privacy-lawsuit, *Google to destroy billions of private browsing records to settle lawsuit* (2024).

[6] The New York Times. https://www.nytimes.com/2022/08/19/technology/tiktok-browser-tracking.html, *TikTok Browser Can Track Users' Keystrokes, According to New Research* (2022).

[7] Wikipedia. https://en.wikipedia.org/wiki/October_7_attacks, *October 7 attaks* (2025).

[8] Amnesty International. https://www.amnesty.org/en/documents/mde15/6701/2023/en/, *Israel and Occupied Palestinian Territories: Automated Apartheid: How facial recognition fragments, segregates and controls Palestinians in the OPT* (2023).

[9] +972 Magazine. https://www.972mag.com/lavender-ai-israeli-army-gaza/, *'Lavender': The AI machine directing Israel's bombing spree in Gaza* (2024).

[10] +972 Magazine. https://www.972mag.com/israeli-intelligence-chatgpt-8200-surveillance-ai/, *Israel developing ChatGPT-like tool that weaponizes surveillance of Palestinians* (2025).

[11] The European Parliament and the Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (2016).

[12] Wikipedia. https://en.wikipedia.org/wiki/Brussels_effect, *Brusselles Effect* (2025).

[13] Richie Koch. https://gdpr.eu/gdpr-vs-lgpd/, *What is the LGPD? Brazil's version of the GDPR* (2019).

[14] Thomas Stoesser. https://insights.comforte.com/canadas-new-data-privacy-bill-the-digital-charter-implementation-act, *Canada's New Data Privacy Bill: the Digital Charter Implementation Act* (2020).

[15] Gibson Dunn. https://www.gibsondunn.com/china-passes-the-personal-information-protection-law-to-take-effect-on-november-1/, *China Passes the Personal Information Protection Law, to Take Effect on November 1* (2021).

[16] Bloomberg Law. https://news.bloomberglaw.com/privacy-and-data-security/move-over-ccpa-the-california-privacy-rights-act-gets-the-spotlight-now, *Move Over, CCPA: The California Privacy Rights Act Gets the Spotlight Now* (2020).

[17] Sarah Rippy. https://iapp.org/news/a/virginia-passes-the-consumer-data-protection-act/, *Virginia passes the Consumer Data Protection Act* (2021).

[18] Sarah Rippy. https://iapp.org/news/a/colorado-privacy-act-becomes-law/, *Colorado Privacy Act becomes law* (2021).

[19] Kiran Garimella, Orestis Kostakis, and Michael Mathioudakis. Ad-blocking: A Study on Performance, Privacy and Counter-measures. page 259–262 (2017). doi:10.1145/3091478.3091514.

[20] Julian Fietkau, Kashyap Thimmaraju, Felix Kybranz, Sebastian Neef, and Jean-Pierre Seifert. The Elephant in the Background: A Quantitative Approachto Empower Users Against Web Browser Fingerprinting. pages 167–180 (2021). doi:10.1145/3463676.3485599.

[21] Sebastian Neef. *Uncovering Fingerprinting Networks. An Analysis of In-Browser Tracking using a Behavior-based Approach* (2022). doi:10.48550/arXiv.2210.11300. ArXiv:2210.11300 [cs].

[22] Libor Polčák, Marek Saloň, Giorgio Maone, Radek Hranický, and Michael McMahon. *JShelter: Give Me My Browser Back* (2023). doi:10.48550/arXiv.2204.01392. ArXiv:2204.01392 [cs].

[23] Giorgio Maone. Hardening the Web with NoScript. *login Usenix Mag.*, 34 (2009).

[24] Wikipedia. https://en.wikipedia.org/wiki/Cross-site_scripting, *Cross-site scripting* (2025).

[25] Lorenzo Laudadio, Riccardo Coppola, Antonio Vetro, Marco Torchiano, and Juan Carlos De Martin. MINOS: A tool to analyze HTTP requests for compliance to GDPR (2024).

[26] Lorenzo Laudadio, Antonio Vetrò, Riccardo Coppola, Juan Carlos De Martin, and Marco Torchiano. Personal Data Transfers to Non-EEA Domains: A Tool for Citizens and An Analysis on Italian Public Administration Websites. *arXiv.org* (2024). ISSN 2331-8422.

[27] Sebastian Deterding, Rilla Khaled, Lennart Nacke, and Dan Dixon. Gamification: Toward a definition. *Proceedings of CHI 2011 Workshop Gamification: Using Game Design Elements in Non-Game Contexts*, pages 6–9 (2011).

[28] Aikaterini-Georgia Mavroeidi, Angeliki Kitsiou, and Christos Kalloniatis. The Role of Gamification in Privacy Protection and User Engagement. In *Security and Privacy From a Legal, Ethical, and Technical Perspective*. IntechOpen (2020). ISBN 978-1-83881-206-5. doi:10.5772/intechopen.91159.

[29] Aikaterini-Georgia Mavroeidi, Angeliki Kitsiou, and Christos Kalloniatis. Gamification: A Necessary Element for Designing Privacy Training Programs. In Christos Kalloniatis, editor, *The Role of Gamification in Software Development Lifecycle*, chapter 1. IntechOpen, Rijeka (2021). doi:10.5772/intechopen.97420.

[30] Daniele Ruggiu, Blok , Vincent, Coenen , Christopher, Kalloniatis , Christos, Kitsiou , Angeliki, Mavroeidi , Aikaterini-Georgia, Milani , Simone, , and Andrea Sitzia. Responsible innovation at work: gamification, public engagement, and privacy by design. *Journal of Responsible Innovation*, 9(3):pages 315 (2022). ISSN 2329-9460. doi:10.1080/23299460.2022.2076985. Publisher: Routledge _eprint: https://doi.org/10.1080/23299460.2022.2076985.

[31] Yu-kai Chou and Erik von Mechelen. *Actionable gamification: beyond points, badges, and leaderboards*. Octalysis Media, Fremont, CA (2016). ISBN 978-1-5117-4404-1.

[32] Wikipedia. https://en.wikipedia.org/wiki/Wikimedia_Foundation, *Wikimedia Foundation* (2025).

[33] Christopher Mims. https://qz.com/115831/googles-20-time-which-brought-you-gmail-and-adsense-is-now-as-good-as-dead, *Google's "20% time", which brought you Gmail and AdSense, is now as good as dead* (2013).

[34] Wikipedia. https://it.wikipedia.org/wiki/Tamagotchi, *Tamagotchi* (2025).

[35] Wikipedia. https://simple.wikipedia.org/wiki/Crying_Indian_public_service_announcement, *Crying Indian public service announcement* (2025).

[36] Youtube. https://youtu.be/3iwHO7WONk0, *1998 Crying Indian "Back By Popular Neglect"* (2007).

[37] Mary-Ann Russon. https://www.ibtimes.co.uk/4bn-mistake-yahoo-bought-precursor-pokemon-go-2013-then-killed-game-1572835, *$4bn mistake: Yahoo bought the precursor to Pokemon Go in 2013 and then killed the game* (2016).

[38] Wikipedia. https://en.wikipedia.org/wiki/Woot, *Woot* (2025).

[39] Daniel Kahneman. *Thinking, fast and slow*. Penguin Books, reissued edition (2024). ISBN 978-0-14-103357-0.

[40] Wikipedia. https://en.wikipedia.org/wiki/FarmVille, *FarmVille* (2025).

[41] Mozilla. https://developer.mozilla.org/en-US/, *MDN Web Docs* (2025).

[42] Statcounter. https://gs.statcounter.com/browser-market-share#monthly-202504-202504-bar, *Browser Market Share Worldwide April 2025* (2025).

[43] Google. https://developer.chrome.com/docs/extensions/reference/api, *API reference* (2021).

[44] Google. https://developer.chrome.com/docs/extensions, *Chrome Extensions Docs* (2025).

[45] Wikipedia. https://en.wikipedia.org/wiki/Google_Chrome#Manifest_V3, *Manifest V3* (2025).

[46] Mozilla. https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model, *Document Object Model (DOM)* (2025).

[47] MonitoraPA. https://monitora-pa.it/, *MonitoraPA* (2025).

[48] Builtwith. https://builtwith.com/, *Find out what websites are Built With* (2025).

[49] European Commision. https://ec.europa.eu/commission/presscorner/detail/en/ip_19_1770, *Antitrust: Commission fines Google €1.49 billion for abusive practices in online advertising* (2019).

[50] The Associated Press. https://apnews.com/article/google-incognito-mode-tracking-lawsuit-settlement-8b30c9397f678bc4c546ab84191f7a9d, *Google settles $5 billion privacy lawsuit over tracking people using incognito mode* (2019).

[51] Politico. https://www.politico.eu/article/google-hit-with-european-privacy-probe-over-its-artificial-intelligence-system/, *Google hit with European privacy probe over its AI system* (2024).

[52] The Guardian. https://www.theguardian.com/technology/2019/jan/21/google-fined-record-44m-by-french-data-protection-watchdog, *Google fined record £44m by French data protection watchdog* (2019).

[53] BBC. GooglefinedÂč91moverad-trackingcookies, *Google fined £91m over ad-tracking cookies* (2019).

[54] Privacy Guides. https://www.privacyguides.org/en/, *Privacy Guides* (2025).

[55] Electronic Frontier Foundation. https://www.eff.org/pages/surveillance-self-defense, *EFF - Surveillance Self-Defense* (2025).

[56] PrivacyTools. https://www.privacytools.io/, *PrivacyTools* (2025).

[57] Daniël Verlaan. https://watchyourhack.com/, *Watch Your Hack* (2021).

[58] Alicia Sykes. https://digital-defense.io/, *Digital Defense* (2024).

[59] LeAlternative. https://www.lealternative.net/, *LeAlternative* (2025).

[60] Etica Digitale. https://privasi.eticadigitale.org/, *PrivaSì* (2025).

[61] _TO * hacklab. https://facciamo.cisti.org/, *Autodifesa digitale* (2025).

[62] Yu-Kai Chou. https://www.yukaichou.com/octalysis-tool/, *Octalysis Tool* (2025).

[63] Mozilla. https://developer.chrome.com/docs/extensions/reference/api, *Javascript APIs - storage* (2025).

[64] Mozilla. https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/runtime, *Javascript APIs - runtime* (2025).

[65] Mozilla. https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/webRequest/onBeforeRequest, *Javascript APIs - webRequest.onBeforeRequest* (2025).

[66] Mozilla. https://developer.mozilla.org/en-US/docs/Web/URI/Reference/Schemes/blob, *blob: URLs* (2025).

[67] Wikipedia. https://it.wikipedia.org/wiki/Media_type, *Media Type* (2025).

[68] Mozilla. https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/downloads, *Javascript APIs - downloads* (2025).

[69] Electronic Frontier Foundation. https://www.eff.org/deeplinks/2021/12/chrome-users-beware-manifest-v3-deceitful-and-threatening, *Chrome Users Beware: Manifest V3 is Deceitful and Threatening* (2021).

[70] Mozilla. https://blog.mozilla.org/en/firefox/firefox-manifest-v3-adblockers/, *Mozilla's approach to Manifest V3: What's different and why it matters for extension users* (2025).

[71] Mozilla. https://extensionworkshop.com/documentation/develop/manifest-v3-migration-guide/, *Manifest V3 migration guide* (2025).

[72] Google. https://developer.chrome.com/docs/extensions/develop/migrate/to-service-workers, *Migrate to a service worker* (2025).

[73] Google. https://developer.chrome.com/docs/extensions/reference/api/offscreen, *chrome.offscreen* (2025).

[74] Giorgio Maone. https://noscript.net/, *NoScript: block scripts and own your browser!* (2025).

[75] Electronic Frontier Foundation. https://privacybadger.org/, *Privacy Badger* (2025).

[76] Raymond Hill. https://ublockorigin.com/, *uBlock Origin - Free, open-source ad content blocker* (2025).

[77] Chromium Blog. https://blog.chromium.org/2024/05/manifest-v2-phase-out-begins.html, *Manifest V2 phase-out begins* (2024).

[78] Raymond Hill. https://github.com/gorhill/uBlock, *uBlock Origin Github Repo* (2025).

[79] Politecnico di Torino. https://www.polito.it/en/education/applying-studying-graduating/choosing-a-degree-programme/polito-open-days-bachelor-s-degree, *PoliTo Open Days Bachelor's Degree* (2025).

[80] Politecnico di Torino. https://www.polito.it/en/education/applying-studying-graduating/choosing-a-degree-programme/polito-open-days-master-s-degree-programmes, *Polito Open Days - Master's degree programmes* (2025).

[81] Wikipedia. https://en.wikipedia.org/wiki/System_usability_scale, *System usability scale* (2025).

[82] Wikipedia. https://en.wikipedia.org/wiki/Likert_scale, *Likert Scale* (2025).

[83] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, page 1245–1254. Association for Computing Machinery, New York, NY, USA (2009). ISBN 9781605584959. doi:10.1145/1557019.1557153.

[84] EasyList. https://easylist.to/, *EasyList* (2025).

[85] Peter Lowe. https://pgl.yoyo.org/adservers/, *Peter Lowe's Blocklistg* (2025).

[86] Ming Di Leom. https://gitlab.com/malware-filter/urlhaus-filter#malicious-url-blocklist, *Online Malicious URL Blocklist* (2025).

[87] DuckDuckGo. https://spreadprivacy.com/duckduckgo-tracker-radar/, *DuckDuckGo Tracker Radar Exposes Hidden Tracking* (2020).