Saarland University

Department of Computer Science

Understanding User Consent Choices: An Exploration of the Runtime Permission Model in the Mobile and Web Ecosystem

Dissertation
zur Erlangung des Grades
der Doktorin der Ingenieurwissenschaften
der Fakultät für Mathematik und Informatik
der Universität des Saarlandes

von
Yusra Elbitar

Saarbrücken, 2025

Tag des Kolloquiums:	25.02.2026
Dekan:	Prof. Roland Speicher

Prüfungsausschuss:

Vorsitzender:	Prof. Thomas Schuster
Berichterstattende:	Dr.-Ing. Sven Bugiel Prof. Dr. Andreas Zeller Prof. Dr. Alena Naiakshina
Akademischer Mitarbeiter:	Dr. Alexi Turcotte

UNIVERSITÄT
DES
SAARLANDES

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Declaration of original authorship

I hereby declare that this dissertation is my own original work except where otherwise indicated. All data or concepts drawn directly or indirectly from other sources have been correctly acknowledged. This dissertation has not been submitted in its present or similar form to any other academic institution either in Germany or abroad for the award of any other degree.

Saarbrücken, 10.09.2025

signed Yusra Elbitar

Zusammenfassung

Moderne mobile und Webanwendungen fordern häufig den Zugriff auf sensible Daten der Nutzer an, um personalisierte und funktionsreiche Erlebnisse zu ermöglichen. Um fundierte Entscheidungen zu unterstützen, bieten Plattformen Entwicklern die Möglichkeit, Berechtigungsanfragen zu kontextualisieren, etwa durch deren zeitliche Platzierung und die Bereitstellung von Begründungen, die erklären, warum eine Berechtigung erforderlich ist. Der Einfluss dieser Strategien auf die Entscheidungen der Nutzer und damit auf ihre Wirksamkeit insgesamt ist jedoch bislang kaum untersucht. Diese Dissertation präsentiert eine umfassende Untersuchung darüber, wie das Design von Berechtigungsanfragen die Entscheidungen der Nutzer beeinflusst. Anhand von drei komplementären Studien, die sowohl mobile als auch Webplattformen abdecken, zeigen wir, wie die Gestaltung von Berechtigungsanfragen durch Entwickler die Wahlentscheidungen der Nutzer prägt. Erstens führen wir eine groß angelegte Nutzerstudie durch, um zu untersuchen, wie das Timing der Anfragen und das Vorhandensein von Begründungen gemeinsam die Entscheidungen und Wahrnehmungen der Nutzer beeinflussen. Zweitens analysieren wir die Sprache und das visuelle Design realer Begründungen in mobilen Apps und identifizieren, wie bestimmte Formulierungen das das Nutzerverhalten steuern. Schließlich erweitern wir unsere Untersuchung auf das Web, indem wir die erste groß angelegte empirische Studie zu Berechtigungsbegründungen in der Praxis durchführen.

Insgesamt liefern diese Studien praxisnahe Erkenntnisse und Gestaltungsempfehlungen für Entwickler und Plattformdesigner. Gleichzeitig werfen sie kritische Fragen zur Sicherung der Integrität und Standardisierung von Berechtigungsbegründungen auf, um Manipulation zu verhindern. Diese Arbeit legt den Grundstein für transparentere, vertrauenswürdigere und nutzerzentrierte Systeme für Berechtigungsanfragen.

Abstract

Modern mobile and web applications frequently request access to users' sensitive data to provide personalized and feature-rich experiences. To support informed decision-making, platforms allow developers to contextualize permission requests by timing them and providing rationales that explain why permission is needed. However, the impact of these strategies on users' decisions, and therefore their overall effectiveness, remains unexplored. This dissertation presents a comprehensive investigation into how the design of permission requests influences users' decisions. Through three complementary studies spanning mobile and web platforms, we show how developers' request permissions influence users' choices. First, we conduct a large-scale user study to examine how the timing of requests and the presence of rationales jointly affect users' decisions and perceptions. Second, we analyze the language and visual design of real-world mobile app rationales, identifying how specific phrasing influences user behavior. Finally, we extend our investigation to the web by conducting the first large-scale empirical study of permission rationales in practice.

Collectively, these studies offer actionable insights and design recommendations for developers and platform designers. They also raise critical questions about ensuring the integrity and standardization of permission rationales to prevent manipulation. This work lays the foundation for more transparent, trustworthy, and user-centered permission request systems.

Background of this Dissertation

This dissertation is based on the papers mentioned in the following. I contributed to all papers as one of the main authors.

Chapter 4 presents the first study conducted as part of this dissertation, published at USENIX Security 2021 [P1]. The research idea came from the author's master thesis, which provided preliminary evidence that the timing of permission requests and the inclusion of justifications for those requests can influence users' permission decisions. However, the master thesis was conducted in a lab environment with a limited number of participants, and the statistical analysis was exploratory in nature, lacking the statistical rigor needed to confirm the findings. To overcome these limitations, the author designed a new study from the ground up, focusing on methodological robustness to address the core research questions. The initial phase focused on understanding the status quo–specifically, how app developers request permissions. The author contributed to the development of a dynamic analysis method for collecting rationales from apps. Trung Tin Nguyen implemented and executed this analysis to gather the raw data. The author then evaluated the collected data and developed a comprehensive codebook to systematically capture the status quo. In addition, the author designed and conducted the user study and carried out the statistical analysis. Michael Schilling provided expertise in empirical research, helping to validate the methodology and offering guidance on data analysis. Sven Bugiel played a key role in strategic decisions and offered ongoing feedback throughout the project. Both Michael Schilling and Sven Bugiel were involved in writing and reviewing the final manuscript.

Chapter 5 presents the second study [P2] of this dissertation, published at NDSS 2025. Building on the author's previous work [P1], which showed that the presence or absence of rationales influences users' permission decisions, this study explores how the phrasing of those rationales affects user behavior. The goal was to identify the most effective ways to communicate permission needs to users through carefully crafted language. The author was solely responsible for developing the methodology to collect both textual and visual components of rationales. This included coding and labeling rationale texts and screenshots, followed by a systematic classification into rationale building blocks. In addition, the author designed and implemented the user study to evaluate the impact of different rationale phrasings on user decisions. Alexander Hart contributed his expertise in empirical research, assisting with the statistical analysis and offering guidance on refining the study design. Sven Bugiel helped shape the project's direction and provided insightful feedback throughout. Both Alexander Hart and Sven Bugiel were involved in the writing process. All authors reviewed and approved the final manuscript.

Chapter 6 is based on the work [P3], presented at CHI 2025. This paper was honored with a CHI Best Paper Award. The author developed the research idea and initiated a collaboration with the Google Chrome team to investigate rationales within a new context—the web ecosystem. Together with Marian Harbach and Soheil Khodayari, the author developed the methodology for collecting rationales from webpages. Soheil Khodayari implemented the web crawler used to gather raw text snippets. The author also worked closely with Soheil Khodayari and Gianluca De Stefano to develop a classifier capable of identifying rationale texts within the collected data, with Gianluca De Stefano

contributing with his machine learning expertise. The author and Soheil Khodayari collaboratively vetted and classified the rationale texts and collected representative screenshots of identified rationales. They also co-developed a comprehensive codebook to capture the diversity of rationale types found on the web. In addition, the author analyzed the visual and design elements of the rationales, categorizing them according to relevant design attributes. Together with Marian Harbach, the author designed an exploratory methodology to examine the influence of rationale presentation on user decisions. While Marian Harbach conducted the statistical analysis, the interpretation of the results and the drawing of conclusions were done collaboratively by the author, Marian Harbach, and Soheil Khodayari. The author and Soheil Khodayari shared equal responsibility for writing the paper. Soheil Khodayari, Marian Harbach, Sven Bugiel, and Balazs Csaba Engedy reviewed and approved the final manuscript.

[P1] **Elbitar, Y.**, Schilling, M., Nguyen, T. T., Backes, M., and Bugiel, S. Explanation beats context: the effect of timing & rationales on users' runtime permission decisions. In: *Proc. 30th USENIX Security Symposium (SEC'21)*. 2021.

[P2] **Elbitar, Y.**, Hart, A., and Bugiel, S. The power of words: a comprehensive analysis of rationales and their effects on users' permission decisions. In: *32nd Annual Network and Distributed System Security Symposium, (NDSS'25)*. 2025.

[P3] **Elbitar, Y.**, Khodayari, S., Harbach, M., De Stefano, G., Engedy, B. C., Pellegrino, G., and Bugiel, S. Permission rationales in the web ecosystem: an exploration of rationale text and design patterns. In: *Conference on Human Factors in Computing Systems (CHI'25)*. 2025.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Ing. Sven Bugiel. I am profoundly thankful for the opportunity he gave me to pursue a PhD under his supervision. Taking up a part-time doctoral position over many years was a significant commitment, and throughout this entire journey, he never questioned my progress or doubted my ability to succeed. His patience, his constant support, and his calm guidance through every obstacle I encountered in my research meant more to me than I can adequately express. His valuable feedback, trust, and encouragement shaped not only this dissertation but also my growth as a researcher. I cannot thank him enough for the opportunity and confidence he placed in me.

I would also like to thank my family, without whom this journey would not have been possible. To my husband, thank you for never doubting me and for standing by my side with unwavering belief and support throughout these years. Your confidence in me gave me strength, especially during the most challenging phases.

To my children, thank you for your patience and understanding, particularly during times when deadlines meant I could not read to you in the evenings or spend as much time with you as I wished. Your love and joy have always been my greatest motivation.

To my mother, father, sisters, and brother, I am deeply grateful for always being there whenever I needed you. Your support went far beyond words. It was present in the countless everyday moments: stepping in to care for the children, taking them out so I could have a quiet hour to think, or simply reassuring me that everything at home was taken care of so I could fully focus on my research. You created the space that allowed me to keep going when things felt overwhelming. You carried responsibilities so I could carry this work forward. Knowing that I could rely on you gave me not only time, but also peace of mind, something that is invaluable during such a demanding journey. This achievement rests, in many ways, on your generosity, your patience, and your constant willingness to support me. Without you, this path would have been far more difficult, if not impossible.

Contents

List of Figures

List of Tables

1 Introduction

In today's digital environment, mobile applications and modern websites increasingly offer feature-rich functionalities and personalized experiences by routinely requesting access to sensitive user data, such as location, camera, or contacts. Each of these requests places a small but significant decision-making burden on the user, who must quickly determine whether to grant or deny access to the requested permission. These decisions, repeated across daily interactions, accumulate into meaningful consequences for user privacy. Yet, users are often left to make these choices on their own with limited information and context.

To assist users and support informed decision-making, developers can contextualize permission requests in two key ways: timing and rationales. On one hand, *Timing* involves aligning the request with user expectations, such as prompting for location access only after the user initiates a "Find stores near me" action, making the request feel natural and purposeful. *Rationales*, on the other hand, are short textual explanations that clarify why an app or site needs a specific permission, helping users better understand the request and make more informed decisions. Although guidelines encourage the use of these techniques [117, 11, 66, 106], developers retain broad freedom in whether, when, and how to apply them. As a result, the actual influence of timing and rationales on user consent remains insufficiently understood.

This dissertation addresses this gap by investigating how developers contextualize permission requests and how these contextual cues influence users' permission decisions across platforms. It presents three interconnected studies: (1) a systematic exploration of the interplay between timing and rationales for mobile permission requests, (2) an analysis of the language and design of rationales and effects of phrasing variations on users' decision-making, and (3) a large-scale study of how permission rationales are used across the web ecosystem. Together, these studies offer new insights into how timing, phrasing, and design impact user permission decisions and provide evidence-based guidance for developers to request permissions across different platforms.

Summary of contributions

Chapter 4: Study on Timing & Rationales of Permission Requests: Current mobile platforms leave it up to the app developer to decide when to request permissions *(timing)* and whether to provide in-context explanations why and how users' private data are accessed *(rationales)*. Given these liberties, it is important to understand how developers should use timing and rationales to effectively assist users in their permission decisions. While guidelines and recommendations for developers exist, no study has systematically investigated the actual influence of timing, rationales, and their combinations on users' decision-making process. In this dissertation, we conducted a comparative online study with 473 participants who were asked to interact with mockup apps drawn from a pool of 120 variations of 30 apps. The study design was guided by developers' current permission request practices derived from a dynamic analysis of the top apps on *Google Play*. Our results show that there is a clear interplay between timing and rationales on users' permission decisions and the evaluation of their decisions, making the effect of rationales stronger when shown upfront and limiting the effect of timing when rationales are present. We therefore suggest adaptation to the available guidelines. We also find

that permission decisions depend on the individuality of users, indicating that there is no one-fits-all permission request strategy, upon we suggest better individual support and outline one possible solution.

Chapter 5: Study on Phrasing of Permission Request Rationales: Rationales offer a method for app developers to convey their permission needs to users. Developers have the creative freedom to design and phrase these rationales, which can significantly influence how users perceive and respond to permission requests. In this dissertation, we explore the characteristics of real-world rationales and how their building blocks affect users' permission decisions and their evaluation of those decisions. Through an analysis of 720 sentences and 428 screenshots of rationales from the top apps of Google Play, we identify the various phrasing and design elements of rationales. Subsequently, in a user study involving 960 participants, we explore how different combinations of phrasings impact users' permission decision-making process. By aligning our insights with established recommendations, we offer actionable guidelines for developers, aiming to make rationales a usable security instrument for users.

Chapter 6: Exploration of Rationales in the Web: Modern web applications use features like camera and geolocation for personalized experiences, requiring user permission via browser prompts. To explain these requests, apps provide rationales–contextual information on why permissions are needed. Despite their importance, little is known about how often rationales appear on the web or their influence on user decisions.

This dissertation presents the first large-scale study of how the web ecosystem handles permission rationales, covering three areas: (i) identifying webpages that use permissions, (ii) detecting and classifying permission rationales, and (iii) analyzing their attributes to understand their impact on user decisions. We examined over 770K webpages from Chrome telemetry, finding 3.6K unique rationale texts and 749 rationale UIs across 85K pages. We extracted key rationale attributes and assessed their effect on user behavior by cross-referencing them with Chrome telemetry data. Our findings reveal nine key insights, providing the first evidence of how different rationales affect user decisions on the web.

Outline The remainder of this dissertation is organized as follows: Chapter 2 provides background information on permission requests across the web and android platform. Chapter 3 reviews related work on mobile and web permission systems, with a focus on user understanding, communication mechanisms, and design strategies. Chapter 4 presents our study on the timing and rationales of runtime permission requests. In Chapter 5, we investigate the phrasing of these rationales, followed by an exploration of web-based permission rationales in Chapter 6. Finally, we draw conclusions and summarize key findings in Chapter 7.

2
Background

Modern applications, whether mobile or web-based, increasingly rely on access to sensitive device capabilities and personal user data such as the camera, microphone, location, contacts, and notification access to deliver personalized and interactive functionality. To protect user privacy and security, both mobile operating systems and web browsers implement permission models that require applications to obtain explicit user consent before accessing such resources. This section provides an overview of permission mechanisms in both the mobile and web domains.

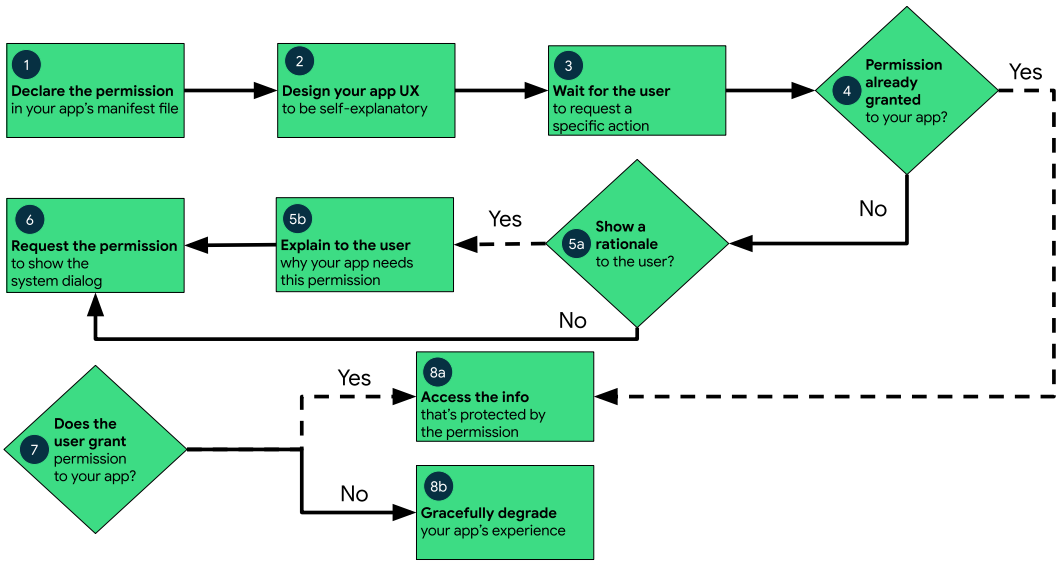


Figure 2.1: Diagram that shows the workflow for declaring and requesting runtime permissions on Android (66).

2.1 Permissions in the Mobile Domain

In the Android ecosystem, apps operate within a sandboxed environment and need explicit permissions to access features such as the camera, contacts, or geolocation. Before Android 6.0 which was released in October 2015, permissions were requested at installation time, forcing users to either accept all permissions or cancel the installation.

With the introduction of the runtime permission model in Android 6.0, permission handling changed significantly. Dangerous permissions, which protect sensitive data, are now requested at runtime when the app actually needs them. Under this model, permissions that are logically related are grouped into a single permission group. For example, both read and write access to contacts are included in the `CONTACTS` group. When requesting permission, the user sees only the name of this permission group.

To use a permission of this type, an app must follow a series of steps, which are illustrated in Figure 2.1. Before requesting permission, an app must first declare the permission in its manifest (Step 1), ensure that the interface clearly explains why the permission is needed (Step 2), and wait until the user initiates an action that requires it (Step 3). After these steps, the app checks whether the required permission has already

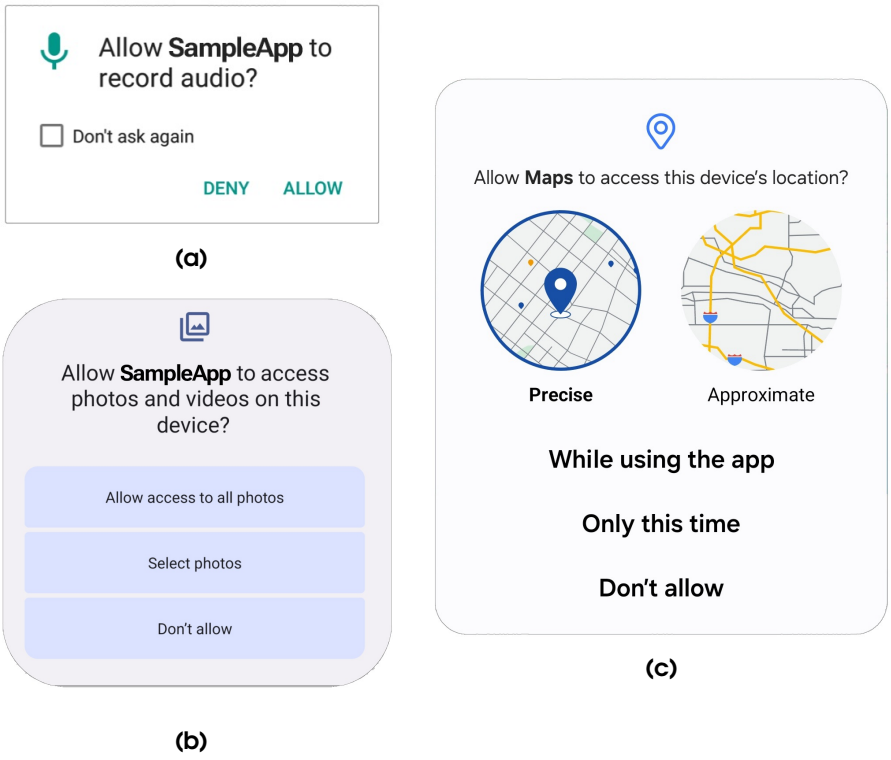


Figure 2.2: Runtime permission request dialogs in different Android versions. (a) Starting in Android 6.0, microphone permission request dialog shows a "don't ask again" button after the user previously denied the permission request. (b) Starting in Android 14.0, storage permission request dialog shows the option to grant access to selected photos. (c) Starting in Android 10.0, location permission request dialog of the `Maps` app shows more fine-grained decision options.

been granted by calling `checkSelfPermission()` [60] (Step 4). If the permission is already granted, the app can proceed without showing a request dialog. Otherwise, it must explicitly request the permission by invoking `requestPermissions()` [67], which triggers a system dialog asking the user to allow or deny the request (Step 6). Figure 2.2 illustrates different types of system permission dialogs introduced in various Android versions. The user's response (Step 7) is then returned back to the app through the `onRequestPermissionsResult()` callback method [65], allowing the app to respond appropriately based on the outcome (Steps 8a and 8b).

To further improve user experience and transparency, Android provides the `shouldShowRequestPermissionRationale()` method [59]. This method allows the app to determine if it should display a rationale before making the request again, for example if the user previously denied the requested permission (Steps 5a and 5b). A rationale is a message associated with a runtime permission request. For instance, the message *"We need Storage permission to save images and documents."* can be presented in a separate UI element, which developers can design as they like (see Figure 2.3a). By briefly explaining why the permission is needed, the app can help the user understand its significance and make an informed decision.

2.1.1 Android Permission Dialog Changes Over Time

Since the introduction of runtime permissions in Android, the permission request dialog has evolved to offer users more granular control over their data. From Android 6.0 through all versions of Android 9, the standard runtime permission dialog appears as shown in Figure 2.2a. If a user denies a permission request, a "Don't ask again" option appears upon the next request. With Android 10, the location permission dialog was updated to include a new option to allow permission only while using the app, giving users more control over when their location is shared (Figure 2.2c). Android 11.0 expanded on this by introducing one-time permissions, allowing users to grant access "Only this time." Android 12.0 continued to enhance privacy by allowing users to choose between precise and approximate location. Starting with Android 14.0, users can grant apps access to only selected photos and videos, offering even finer control over media sharing, as depicted in Figure 2.2b.

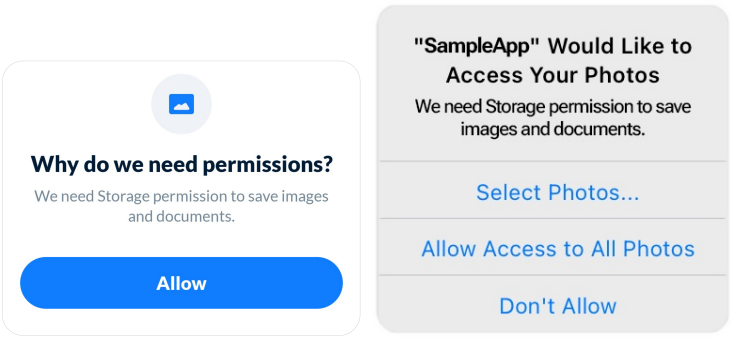


Figure 2.3: Example rationales in the mobile domain. (a) Example of a permission rationale shown on a custom dialog of the `PDF-Scanner-App` on Android. (b) The same rationale message is shown on the permission request dialog on iOS.

2.1.2 Runtime Permissions in iOS

The runtime permission model is not exclusive to Android but is also employed by other mobile platforms, such as iOS. Like Android, iOS requires apps to request permissions to access sensitive user data and device features. However, when it comes to providing rationales, iOS developers must include a static purpose string in the app's `Info.plist` file [14]. This string is automatically displayed in the system permission dialog and must clearly explain why the permission is needed [15], as shown in Figure 2.3b. As with Android, developers may also present optional pre-permission dialogs (also known as pre-alert screens) to provide additional context before triggering the system prompt [12]. This would be similar to the example rationale on Android in Figure 2.3a

The runtime permission model gives developers control over both the timing and the inclusion of rationales, creating a channel of communication between the app and the user. Permission requests can be triggered either at the start of the app or in context when access to a permission-protected resource is needed, and rationales can be

added to assist users' in their decision. Understanding how the timing and presence of rationales impact user decisions is critical for improving how permissions are requested and enhancing the overall user experience.

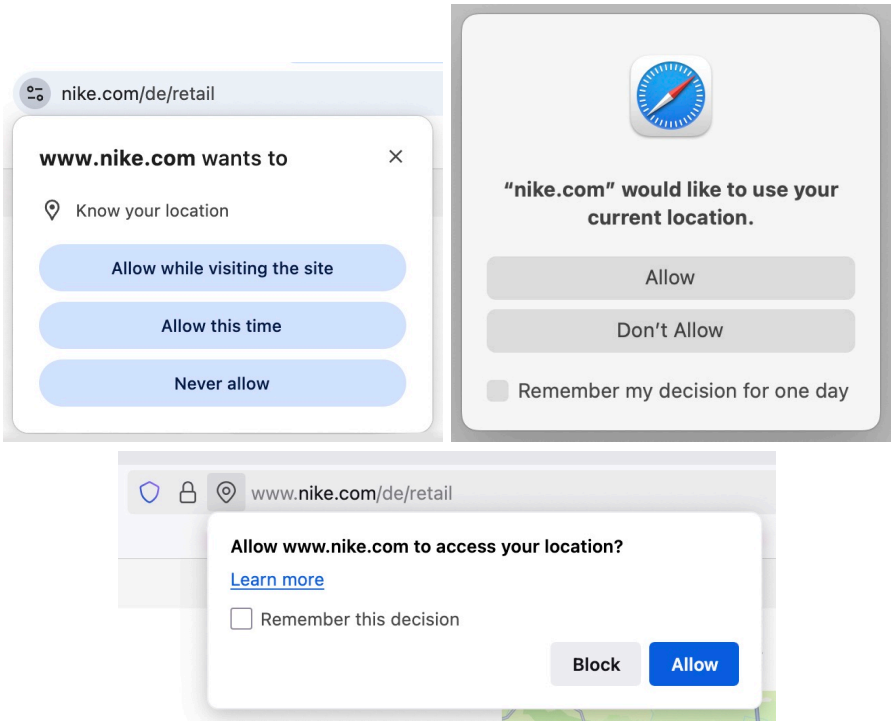


Figure 2.4: Examples of permission request prompts displayed by `nike.com` across different web browsers. *Top left:* Chrome. *Top right:* Safari. *Bottom:* Firefox.

2.2 Permissions in the Web Domain

Similarly, modern web applications increasingly depend on permission-protected Web APIs to access device capabilities and personal user data, which enable rich, interactive experiences on the web without the need for native applications. Access to sensitive resources such as the camera, microphone, location, and notification systems is mediated by browser permission prompts, which serve as a runtime user consent mechanism to enforce the principle of least privilege and mitigate privacy and security risks. Examples of such APIs include: 1) The `MediaDevices` API enables access to audio and video input devices, typically for video conferencing, audio recording, or augmented reality (AR) applications [161]. 2) The `Geolocation` API provides coarse or fine-grained location data, often used in mapping, weather forecasting, or location-based personalization [113]. 3) The `Notifications` API and the `Push` API support asynchronous communication with the user by allowing web apps to send system-level notifications, even when the site is not actively loaded in a tab [169, 162].

These APIs are protected by permissions, so the browser prompts the user for consent the first time a webpage tries to access them. These prompts appear near the

address bar and are usually non-modal, i.e., they do not block further interaction with the website. Users have the option to *grant*, *deny*, or *ignore* these permission prompts. Many popular browsers also offer a temporary block option. In Chrome and Edge, this is done by clicking the *"X"* to dismiss the prompt, while in Firefox, users can click *"Block"* without selecting *"Remember this decision."* In this dissertation, we refer to this action as *dismiss*, consistent with terminology used in prior research [71]. Figure 2.4 shows the location permission prompt as it appears in different browsers.

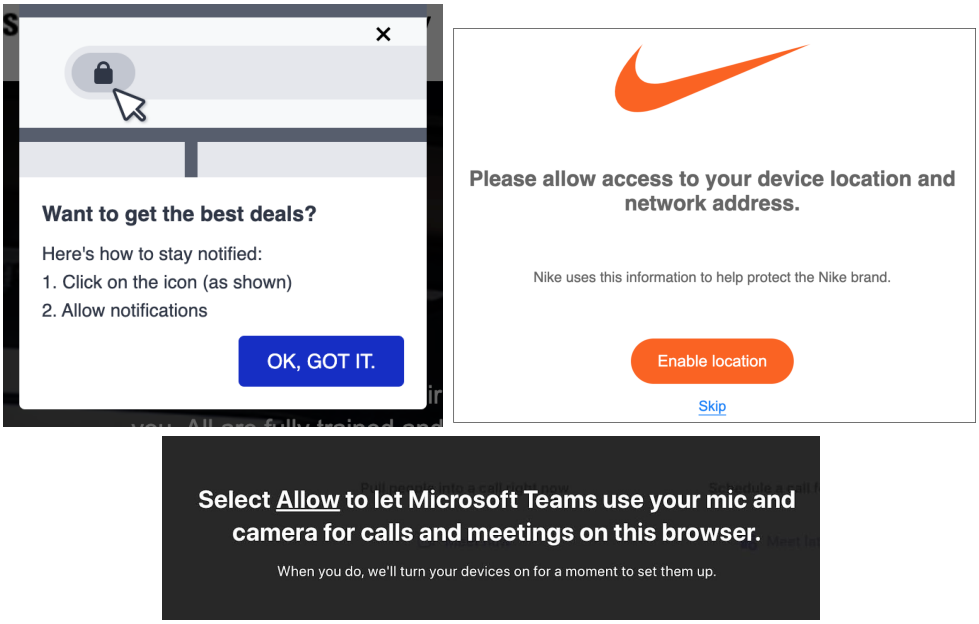


Figure 2.5: Examples of permission rationales in real websites. *Top Left:* notification rationale in `samsung.com`. *Top Right:* geolocation rationale in `nike.com`. *Bottom:* camera and microphone rationale in `teams.microsoft.com`.

Websites, like Android apps, can offer rationales, explanatory UI elements that inform users why a particular permission is being requested. These UIs are implemented entirely by developers and may appear before, alongside, or after the browser's permission prompt (see Figure 2.5). However, unlike Android, where the `shouldShowRequestPermissionRationale()` method [59] provides platform-level support for presenting context-aware rationales after a prior denial, the web platform lacks any standardized or API-driven mechanism to detect prior denials or to integrate rationales into the browser's permission workflow. As a result, websites receive no feedback on whether or how often a user has denied a permission request, nor do they have programmatic means to tailor their rationales accordingly. Web developers must therefore rely on heuristics or user interaction patterns (e.g., inferring hesitation from prompt dismissal) and cannot distinguish between different types of denials. Moreover, since browser permission prompts are typically non-modal and transient, users may easily miss or ignore rationales unless they are clearly and prominently presented within the page's interface. In this dissertation, we explore the current state of web permission rationales, including their types, components, and UI design choices.

3
Related Work

Access to protected resources or sensitive user data is often needed for mobile and web applications to deliver their intended functionality. To access such data, developers must request permission, and users are expected to make informed decisions about whether to grant or deny these requests. However, previous research, particularly in the mobile domain, has shown that users often struggle to understand what permissions are being requested, why they are needed, and what risks they may entail [86, 48, 127, 97]. This disconnect between developer intention and user comprehension can lead to confusion, mistrust, and unmet expectations [23, 170, 24, 86, 48, 28, 155].

To address this challenge, researchers have explored a range of tools and strategies aimed at improving the clarity of permission requests, ultimately helping users make more informed decisions. In the following sections, we first review prior work in the mobile domain that focuses on enhancing user understanding of permissions. We then examine research on the timing and rationales of permission requests, which can help provide contextual cues to users. Finally, we consider related efforts in the web domain, where similar challenges around permission transparency have been explored.

3.1 Mobile Permissions

Various efforts aim to help developers make permission requests more transparent and user-friendly. For instance, some tools support the transition from the install-time permission model to the runtime model [55, 22]. Other research provides best practices for minimizing user burden when requesting permissions [126]. These include strategies such as deferring non-essential permission prompts until the point of use, tying requests to clear user actions, and presenting justifications that are easy to understand and contextually relevant. More recently, automated techniques have emerged that alert developers when a permission request is likely unnecessary by comparing it with the behavior of similar apps [124].

From the user perspective, research has focused on ways to reduce decision-making fatigue by predicting how users are likely to respond to permission requests based on privacy profiles [100, 96, 101, 171, 119], or by providing privacy nudges that guide users toward more informed choices [174, 8]. Additionally, fine-grained permission managers have been proposed to give users greater control over their data [158]. Moreover, previous research suggested that permission requests should be tied directly to user actions within the app, as users are more likely to understand and accept these requests when they occur in response to something they have done [109].

Developers have multiple channels through which they can communicate their apps' use of permissions to users. Traditional methods include app descriptions and privacy policies. Research on app descriptions has introduced techniques for automatically extracting permission-related information from app text or source code [123, 103, 163, 122, 175]. These techniques help assess how accurately app descriptions reflect the actual use of permissions [132, 49, 52], a task that becomes especially important when permissions are requested by embedded third-party libraries—code that developers may not fully understand or control [143, 121, 150, 167].

More recent tools such as Apple's Privacy Nutrition Labels [13] and Google Play's Data Safety Section [68] aim to present privacy practices in a more digestible format.

These tools are designed to simplify complex privacy policies by providing concise, standardized summaries. However, studies have shown that these approaches still pose challenges for users and developers, who often struggle to interpret them accurately [177, 140, 176, 88, 92, 33, 136, 94]. These limitations have prompted further research into how such tools can be made more understandable and effective [98, 176, 177, 33].

3.1.1 Permission Rationales

While the above methods offer asynchronous means of conveying information, the most direct way to explain permission needs is at runtime through in-app rationales. These messages appear around the permission request and can significantly influence user decisions. Several general guidelines exist for designing and phrasing rationales [117, 11, 66], recommending clarity, brevity, and user-centered framing. For example, Nielsen Norman Group emphasizes using plain language and making the benefit to the user explicit [117], while Apple and Google offer platform-specific recommendations for providing rationales (or purpose strings in case of iOS) that align with the permission context and user expectations [11, 66].

In addition to these platform-provided recommendations, research has suggested further best practices, such as ensuring transparency, avoiding technical jargon, and clearly articulating how the requested permission relates to core app functionality [86, 126]. Despite the availability of such guidance, developers often underutilize this mechanism. When rationales are included, they are frequently vague, uninformative, or poorly tailored to user concerns [102, 156, 96]. Furthermore, the voluntary nature of rationale presentation and the absence of enforced standards mean that implementation varies widely across applications. This raises two important questions: how developers currently phrase and design rationales, and how variations in their wording affect users' decision-making behavior.

Limited research has explored the influence of rationales on user decision-making. Previous studies have shown that users benefit from additional information in permission requests [142, 165, 174]. Specifically, earlier research has indicated that including rationales increases the likelihood of users granting permissions [156, 28]. Furthermore, similar findings exist in other domains. For instance, in the field of two-factor authentication (2FA), personalized messages have been found to enhance user adoption of 2FA [58]. Additionally, the terminology used in app descriptions shaped user perceptions of security within secure messaging apps [5, 37]. However, it is crucial to emphasize that the specific effects of rationales in the mobile domain have yet to be investigated.

While the influence of rationale phrasing on users' permission decision-making has received limited attention, the significance of language in decision contexts is well-established in various fields. One such phenomenon is the framing effect [159], a concept extensively studied in psychology [149]. The framing effect reveals how decisions and judgments can be swayed by how information is presented or framed [159, 90]. Other studies have shown that subtle wording differences can significantly influence customers' attitudes toward brands [141]. By drawing connections between these diverse findings, it is plausible that variations in how permission rationales are phrased could similarly affect users' decisions, even without changing the core information provided.

3.1.2 Timing of Permission Requests

Beyond providing static information about data practices, the timing and context in which permission requests are made significantly influence users' decisions. Studies highlight the importance of contextualizing permission requests, where developers not only provide rationales but also determine when to prompt users for permissions [157, 160, 28]. Users are more inclined to grant permissions when the request is made at an appropriate moment, and the purpose and recipient of the data are clearly communicated. This underlines the importance of well-timed and well-explained permission requests to assist users in making informed decisions. Furthermore, it raises key questions about when developers ask for permission and how the timing impacts user behavior and decision-making process.

3.2 Web Permissions

Shifting to web permissions, research has been limited, with most studies focusing on permission prompts and APIs [31] rather than rationales. A significant portion of this work has centered around push notifications, particularly their potential for misuse. Studies have highlighted how unethical content providers exploit these notifications on both mobile and desktop platforms, sending irrelevant or abusive messages to drive traffic [17, 152]. To address these issues, browser vendors such as Firefox [114], Edge [110], and Chrome [20] have implemented features to minimize unwanted interruptions. For example, Chrome experiments showed that quieter prompts, which are less visually prominent, significantly reduce interruptions while maintaining similar grant rates [20].

In addressing the issue of frequent and potentially disruptive permission prompts, recent work [72] introduced a machine learning-based system in Chrome designed to predict when users are unlikely to grant permissions, thereby suppressing prompts deemed unnecessary. While this approach effectively reduces interruptions, it primarily focuses on whether a prompt should be shown, based on predicted user behavior. In contrast, we focus on how permission requests, once displayed, can be communicated more effectively to users. Other research [76] explored vulnerabilities in web permission prompts, such as click-jacking attacks targeting webcam access, highlighting the need for stronger user protections.

Finally, recent research [71] studied user interactions with web permissions in a large-scale analysis of 100 million Chrome installations. This research found that geolocation and notification prompts are often dismissed or ignored, while contextual information significantly increases users' likelihood of granting permissions. However, no research has yet thoroughly explored the use of rationales for web permissions, particularly on desktop platforms, marking a gap in our understanding of how to best support users in making informed permission decisions.

4
Timing & Rationales

Explanation Beats Context: The Effect of Timing & Rationales on Users' Runtime Permission Decisions

4.1 Motivation

Mobile platforms such as Android and iOS handle some of users' most private data, can precisely record information using available sensors, and are "always on". To keep users in control, these platforms make it possible for users to delegate access rights (permissions) to apps. As such, the user decides *which* app is granted which permissions, while it is up to the app developer to decide *when* to ask the user for permission and whether to provide an explanation as to *why and how* data is accessed. The timing of permission requests, along with the accompanying explanations or "rationales", form a one-way communication channel from developers to users. This channel conveys information meant to help users make informed permission decisions which reflect their individual values and privacy preferences in a given context.

Prior work [126] as well as current Google guidelines [106] contain recommendations for developers about when and how permissions should be requested. Although the available advice seems straightforward, there is not enough scientific evidence to thoroughly support it. We unfortunately do not know how timing, rationales, and their combinations affect users' decisions, which strategies in asking for permissions help users the most, and whether those guidelines agree with users' preferences. In the literature, a large body of work has focused on understanding the reasons behind users' permission decisions [24, 170, 100, 96, 97, 143, 174], but all those prior studies have been conducted either on the obsolete install-time permission model or on the current permission model but without considering the different variations depending on timing and rationales within the model itself. Other researchers studied the isolated effect of rationales on users' permission decisions [156] or developers' current rationale practices [102, 156]. Prior works that considered both timing and rationales only reported the status quo of developers' current permission request practices [55]. This leaves a gap in the understanding of the effects and interactions of these variables on users' decisions and whether these decisions mirror the individual interests of users.

4.2 Contribution

In this work, we investigate how two key factors influence users' perception of their decision-making process when responding to permission requests. These factors are the timing of the request (upfront vs. in-context) and the inclusion of rationales (present vs. absent). We also explore how developers can use these elements to better support users in making informed choices about granting permissions. To address these questions, we conducted what we believe is the first comprehensive analysis of the communication channel for runtime permission requests from both the developer's and the user's perspectives. We began with an empirical study by dynamically analyzing top apps on the Google Play Store to examine how developers request permissions during app use (Section 4.3). Through this process, we collected over 2,5K dangerous permission requests, providing insight into current developer practices.

Based on these findings, we designed a user study with 473 participants recruited through Amazon MTurk. The purpose of the study was to evaluate the effects of timing, the inclusion or exclusion of rationales, and their interactions on users' permission

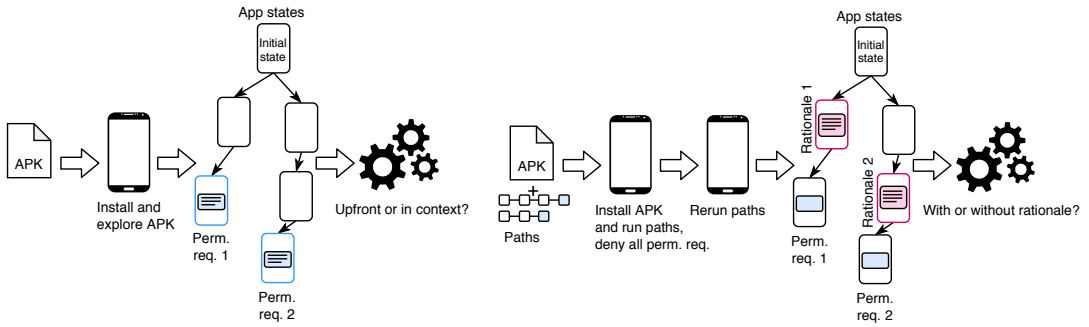
decisions (Section 4.4). We used a standardized rationale format, informed by our empirical analysis, to ensure consistency. To simulate realistic conditions, we created 30 interactive mockup apps inspired by real-world applications. Each app was implemented in four versions, representing the possible combinations of timing and rationale inclusion: upfront with rationale, upfront without rationale, in context with rationale, and in context without rationale. Participants responded to approximately 1,8K permission requests during the study. We collected data on their decisions, their perception of having made an informed choice, their satisfaction with the decision, their sense of control, and how well they understood the purpose of the requested permission.

Our results (Section 4.5) indicate a mutual interplay between the timing of permission requests and rationales. Overall, we found that rationales increase grant rates and have a positive effect on users' perception of their decisions. However, this effect is stronger when rationales are added upfront rather than in context. As for timing, on one hand, asking for permissions in context has a positive effect on users' perception when no rationales are present. On the other hand, requesting permissions in context always has a positive effect on grant rates, regardless of the presence of rationales. Based on these findings (Section 4.6), we suggest the adaptation of Google's current guidelines [106] to better support users in their decision-making process. Going beyond these aspects, however, we also found that permission decisions depend on individual differences between users. As a consequence, we argue that there is no one-size-fits-all permission request strategy. Therefore, current mobile platforms could benefit from built-in support for users to customize permission requests. This could be realised through a system setting that would enable users to configure when they would like to see permission requests and whether they prefer to see rationales.

4.3 Empirical Analysis

We conducted an empirical analysis of rationales and timing of permission requests in the top apps from Google Play. The main goal of this analysis was to provide a valid foundation for the standardized rationale design and select the apps for the user study (see Sections 4.4.5 and 4.4.6 for more details). Our crawler collected the top 100 free apps in each category from Play (Dec. 2018–June 2019). We expected to find a representative sample of apps using runtime permission requests, since we conducted the analysis three years after the runtime permission model was introduced (with the release of Android 6 in Oct. 2015) and one month after this model became mandatory for all new apps and app updates [35]. The top 100 apps varied during the 7-months long crawling period. We therefore collected more than 200,000 unique apps.

Our initial approach to detect timing of permission requests and rationales was to use static analysis. However, we discovered that this approach cannot provide reliable information about the exact position of permission requests in the GUI control-flow. Thus, we used static analysis only to reduce the number of apps that will be subjected to dynamic analysis by filtering out all apps that do not request dangerous permissions in their manifest and do not call the `requestPermissions()` API. We also removed non-English as well as game-related apps. From the resulting set of 12,794 apps, we then randomly selected 10,000 apps for further analysis.



(a) Step 1 identifies timing of permission requests: Heuristics are applied on extracted paths, consisting of a list of app-states from start to each permission request.

(b) Step 2 identifies rationales: Extracted paths are rerun twice, in the first run all permissions are denied, in the second run rationales are extracted.

Figure 4.1: Steps of the empirical analysis

4.3.1 Classification of Permission Requests

For the dynamic analysis we extended DroidBot [93], a lightweight test input generator for Android apps. In two analysis steps, we determined the timing of permission requests (step 1) and the presence of rationales (step 2).

Step 1: Identify timing. This step occupied most of the dynamic analysis time (~30–60 min per app). As shown in Figure 4.1a, we first installed and launched the app of interest. Then we waited around 60 seconds before exploring the app. This step was important to correctly identify upfront permission requests that would otherwise have been categorized as in-context because some apps take time to load (e.g., using a splash screen). The output of the dynamic analysis was the shortest path to all permission requests found. Each path consisted of a list of states from app launch to the permission request of interest, on which we applied a set of heuristics to identify the timing. For example, if the permission request appeared without clicking on some UI element, we considered the timing upfront.

Step 2: Identify rationales. To also find rationales that were only displayed after a permission has been denied, we first reinstalled the app, followed each permission request path from step 1, and denied all requests (as shown in Figure 4.1b). Then, we ran each path again and collected the resulting app states, possibly with new rationale messages. To extract these messages, we used rationales that were obtained with a CNN classifier by previous work [102] in a Latent Semantic Analysis (LSA) to group similar rationales under one topic. These topics were then used in a semantic similarity analysis [111] that assigned a score to each sentence in the permission request path. All sentences that were at least 40% similar to a rationale topic were then manually verified as rationales. We used the evaluation of 100 randomly selected permission requests (50 categorized with rationale and 50 without) as a benchmark to evaluate this threshold. The classification of this subset had a precision of 94% and a recall of 100%.

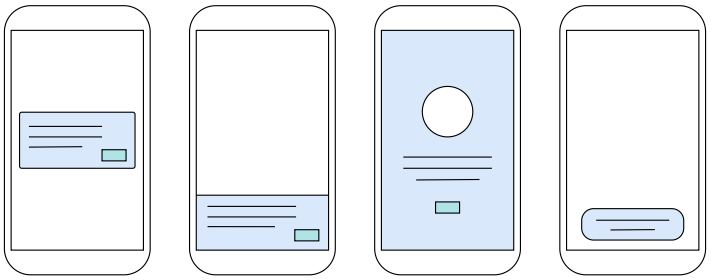


Figure 4.2: The different rationale designs.

From our initial app set, we successfully analyzed 7,998 apps and found 2,071 apps that requested at least one dangerous permission at runtime (total of 2,569 permission requests). Upon closer inspection, we found that part of this discrepancy was due to the fact that many apps included the `requestPermissions()` API in third-party library code that was never executed, what meant that we spent time dynamically analyzing apps that did not actually request permissions at runtime. Further, low code coverage of dynamic analysis (e.g., through login-forms) is a known limitation of available analysis tools, which prevented us from reaching all permission requests. Nevertheless, we collected an adequate number of rationales that were used in the selection process of the standardized rationale for the user study.

4.3.2 Findings

As the results are biased towards upfront permission requests, we should consider them with reservation. Nevertheless, we reveal different ways of showing rationales in terms of design, quality, wording, and timing.

Timing and presence vs. absence of rationales. Of the 2,569 found permission requests, 70% were displayed upfront and 16% showed rationales that were evenly distributed among upfront and in-context requests. The most frequently requested permission was `STORAGE` (56% of 2,569) followed by `LOCATION` (19%), `CAMERA` (9%), `PHONE` (6%), `CONTACTS` (3%), and `MICROPHONE` (3%). We only found a small number of permission requests for `SMS`, `CALENDAR`, and `PHONELOG`, which is consistent with prior work [24, 109]. A chi-square test of independence was performed to examine the relation between timing and permission type. Due to too few observations we excluded the permissions `SMS`, `CALENDAR`, and `PHONELOG` from the analysis. We found that the proportion of in-context permission requests significantly differed between permissions ($X^2(5) = 49.562$, $p < 0.001$, $Cramer's\, V = 0.139$). For example, the highest proportion of in-context requests was found for `STORAGE` (34%), closely followed by `MICROPHONE` (33%), and `CAMERA` (32%). While the lowest proportion was seen for `LOCATION` (23%) and `PHONE` (12%), which are often associated with background functionalities and are therefore most frequently requested upfront. Whereas, there was no significant association between permission type and presence/absence of rationales ($X^2(5) = 8.06$, $p = 0.153$, $Cramer's\, V = 0.056$).

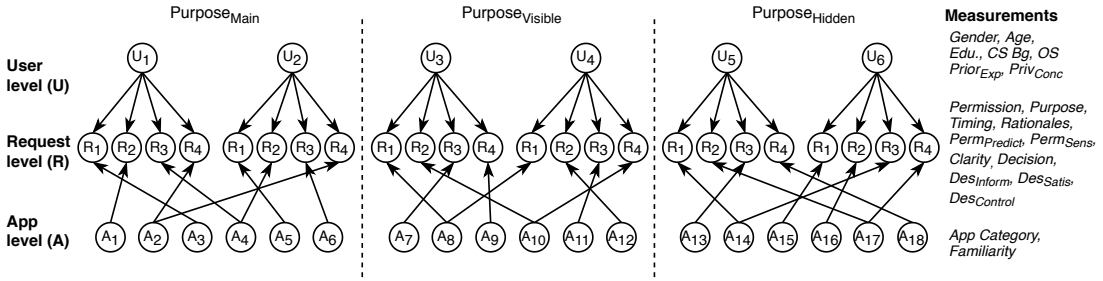


Figure 4.3: Hierarchical structure of the user study.

Design and wording of rationales. We found four general design patterns for rationales. They were displayed as either dialogs, fullscreen views, banners, or snackbars (as highlighted in Figure 4.2). Each design pattern was shown before a permission request or after a permission denial, except for snackbars which were only used after a permission was denied. Additionally, each design provided rationales for one or multiple permissions. We also noticed that most rationales provided an acknowledge button (e.g., ok, got it, proceed), while around half of the dialogs additionally included a cancel button (e.g., cancel, exit, not now, skip). The fullscreen views had the most design variations, compared to the other options, which mostly used the default Android layout.

As for the content, rationales either provided more information compared to the default permission request dialog (i.e., reasons why the app needs the permission and how it will be used) or they just signified that some permission is required or has been denied (e.g., this app requires this permission: to work perfectly, run normally, function properly). We found that about 50% of the rationales provided additional information, thus fulfilling the true purpose of rationales.

4.4 User Study

The aim of this user study is to assess whether there is an effect of timing and presence/absence of rationales on users' permission decisions. To isolate these effects, we used the findings from the empirical analysis to define a standardized rationale that also explains how and why a permission is needed (providing additional information). More precisely, we want to answer the following questions: How does the interaction of timing and presence/absence of rationales affect (1) users' runtime permission decision, (2) the evaluation of their decision, and (3) their perceived clarity of the permission purpose? Since timing and rationales differentiate the runtime permission request model from its predecessor, it is essential to understand how these factors affect users from different perspectives, even after considering other key factors found in prior work. By answering this question, we expect to gain insights on how developers should request permissions to maximize the benefits of the runtime permission model. Based on these findings, we will also discuss Google's guidelines [106] and potential system support.

For a holistic understanding of user's perspective, we included both the permission decision (grant/deny) and the subjective evaluation of this decision as outcome variables, where the latter reflects whether the decision was made according to users' individual

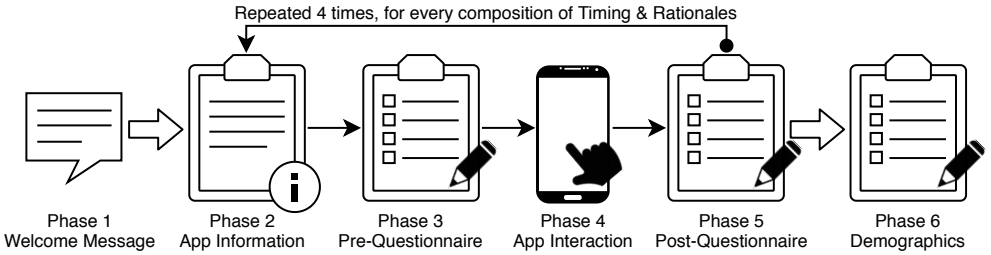


Figure 4.4: Study procedure. *Timing = upfront vs. in-context, Rationales = present vs. absent.*

privacy preferences in a given context. For this, we used the Decision Evaluation Scales (DES) [148] which we adopted from the field of health psychology. These scales were originally designed to evaluate patients' decision to uptake/refuse a treatment choice. Comparing users' permission decision with patients' treatment choice, both have two options: grant/deny a permission or uptake/refuse a treatment. Additionally, both have a direct impact on users' security or patients' health. Based on these similarities, this measure fits the context of our study, especially considering that the DES account for the multidimensional nature of decisions and capture (1) whether users received sufficient information to make an informed decision, (2) their satisfaction with the decision, and (3) their perceived control over the decision. We also measure users' understanding of why the app needs the requested permission, which provides information about how certain combinations of timing (upfront vs. in-context) and rationales (present vs. absent) better communicate permission purposes.

4.4.1 Study Design

We designed the study as an online experiment with repeated measures. Experimental research has the unique strength of high internal validity because it is able to isolate causal relationships through systematic manipulation of the variables of interest (timing and presence/absence of rationales) while controlling for the spurious effect of other extraneous variables (user and app-related differences) [18, 145]. We used a within-subject design (repeated measures) because it reduces errors associated with individual differences and because the alternative (between-subjects) was shown to produce misleading results for studies involving judgment [21]. Since every study design has its limitations, we address these in Section 4.7. To facilitate comparison of user responses, participants were asked about permission requests that shared the same general purpose. These purposes were derived from prior work [100, 96, 97] and include permissions required for the app's core functionality, a visible feature, or a hidden feature. While field studies offer certain advantages, such as high external validity, we opted against conducting this study in the wild due to several drawbacks. For instance, using an app with accessibility features would require continuously logging app changes, raising privacy concerns and introducing opt-in bias. Additionally, we would need to revoke all permissions initially to track participants' decisions and deny alle requests once to present most rationales, forcing participants to navigate a complex workflow.

Our study had a hierarchical structure in which users interacted with permission

requests from different apps. To account for the fact that observations for the same user and app would be similar to each other, we designed this study using a multilevel model [79]. Multilevel models are used for the statistical analysis of hierarchical data, where groups in the study are treated as a random sample from a population of groups. This allows us to make statistical inferences about the population of apps and users, beyond the ones present in the study [79]. Figure 4.3 depicts the levels of the user study. Each user interacted with four permission requests on the *Request* level, one per possible combination of timing (upfront vs. in-context) and rationales (present vs. absent). These permission requests belonged to four different apps and the order of the requests was randomized. The *Request* level captures the characteristics of individual permission requests, including the outcome variables influenced by the experimental conditions, as well as by the type, purpose, predictability, clarity, and sensitivity of the request. The *App* level captures the characteristics of each app, including its category and the participant's familiarity with it. Finally, the *User* level captures the characteristics of each participant, including gender, age, education, computer science background, mobile OS, privacy concerns, and prior privacy experiences.

4.4.2 Procedure

As shown in Figure 4.4, participants first read about the study and gave their consent (phase 1). This was followed by the main part of the study during which participants went through phases 2–5 four times, once per possible composition of timing (upfront vs. in-context) and rationales (present vs. absent), each time for a different app. These phases were designed to come closest to users' interaction with real-life apps. For that, we gave participants a goal to achieve through the app. We also provided participants with the app's description, name, and icon so they had an idea what the app was about. In addition, we used interactive mockup apps, allowing participants to click through the app interactively, just like on their real phones. The user study procedure with a sample mockup app is shown in Appendix 8.1. Phases 2–5 are described next.

Phase 2: App information. Participants were introduced to the app by receiving a brief description of its functionalities, and a goal they needed to achieve through the app (e.g., you want to use this app to have a conference call with your work colleagues, or you want to use this app to backup your vehicle's data). Each goal was based on one of the app's functionalities that would also require a permission. We also provided participants with the app name and icon.

Phase 3: Pre-questionnaire. This phase covers users' first impression of the app. We asked participants whether they were familiar with the app, and if they would expect it to request access to a permission protected resource specific to each app. We also measured the perceived sensitivity of the requested permission (Perm. Sensitivity), and clarity of the permission purpose (Clarity Pre).

Phase 4: App interaction. We reminded participants of the goal they want to achieve through the app and then asked them to interact with an interactive mockup app like

they would on their own phones. Each app interaction ended with a permission request dialog. The order in which participants interacted with the different combinations of timing (upfront vs. in-context) and rationales (present vs. absent) was randomized.

Phase 5: Post-questionnaire. After participants interacted with the app, we asked them if they would grant the requested permission (Decision). We again recorded participants' clarity on the permission purpose (Clarity Post). Other questions inquired about the purpose category of the permission request (Purpose), and some questions were only present when rationales were provided. They investigated the origin of the rationale message (Rationale Origin), and their collection of the content of that message (RationaleRecall). Then, on a separate screen, we reminded participants about their previous decision and asked them to evaluate their choice using the Decisions Evaluation Scales (DES), consisting of informed decision (DES Inform), decision satisfaction (DES Satis), and decision control (DES Control).

After answering questions for the four apps, participants were asked to provide some demographic information (phase 6). The study procedure and all measurements were tested and adjusted after running a pilot study with 25 participants.

4.4.3 Recruitment and Incentives

Data were collected on MTurk using TurkPrime [99], an online platform that facilitates setting up and executing studies on MTurk. We paid participants $12.00/hour, meaning that participants received $3.00 for completing this 15 min study.

To ensure high quality of data collected through MTurk, we followed a number of suggestions in the literature [84, 135]. MTurk workers could only participate in the study if they had a US account and had an approval rate of at least 95%. In order to also collect responses from naive workers (i.e., workers who were not repeatedly exposed to similar studies), we set the required number of completed HITs between 0 and 100 for about 10% of all HITs. Additionally, we added the completion code at the beginning of the study (phase 1) to increase participants' trust (only 1.15% tried to submit the completion code without doing the survey). Finally, we provided one attention check item in the middle of the study and monitored whether participants interacted with the mockup apps. We excluded participants who failed the attention check and did not interact with at least two of the mockup apps.

Since power analysis for multilevel models is still considered a complex problem [79], we estimated the required sample size without considering the multilevel structure of our data. Using G*Power [47], we estimate that we need at least 400 participants. A total of 698 MTurk workers attempted to participate in our study, from which we removed 225 respondents based on the screening criteria described above. Our final sample included 473 participants, 36.8% ($N = 174$) of whom identified themselves as female. The mean age was 37.08 years ($SD = 10.59$). The majority of participants attended college, 17.5% did not finish their studies, 51.4% had a bachelor's degree, and 18.4% had a graduate degree. 69.8% owned an Android smartphone, and 28.3% an iPhone. About one third of all participants had a background in computer science. Appendix 8.2 shows the demographics of the sample.

4.4.4 Measurements

We used different measurements in our study, which are described next and are listed in the questionnaire in Appendix 8.1.

Decision Evaluation Scales (DES). We used the Decision Evaluation Scales (DES) [148] to assess users' permission decisions. The DES consists of three subscales: informed decision, decision satisfaction, and decision control. These scales were originally developed to evaluate how patients assess their medical treatment choices. Because medical decisions often involve multiple parties (e.g., doctors and family members), whereas permission decisions are typically made individually, we adapted each subscale accordingly. To do this, we employed an expert rating procedure to select the most suitable items for each subscale. The experts included professionals from computer science ($N = 3$) and psychology ($N = 4$). The phrasing of the DES question was as follows: *"In a previous question you chose to {grant/deny} this app access to your {permission protected resource}. We would like to know how you feel about this decision."*

Informed Decision (DES Inform): This subscale measures whether users feel they have received sufficient information to make a decision and consists of four items ($\alpha = 0.76$). Sample items include *"I made a well-informed choice"* and *"I know the pros and cons of granting this app access to my {permission protected resource}."* Items are rated on a 7-point scale (1 = strongly disagree; 7 = strongly agree), with higher scores indicating a better-informed decision.

Decision Satisfaction (DES Satis): This subscale measures users' overall confidence and satisfaction with their decision. Sample items include *"I am satisfied with my decision"* and *"I am doubtful about my choice"* (reverse coded), with four items in total ($\alpha = 0.84$). Items are rated on a 7-point scale (1 = strongly disagree; 7 = strongly agree), with higher scores indicating greater satisfaction.

Decision Control (DES Control): Measures whether users had the feeling that they were forced to their decision. This scale consists of four items ($\alpha = 0.80$). Sample items are *"I feel that the app forced me to make this decision"* (reverse coded) and *"This was my own decision."* Items are rated on a 7-point scale (1 = strongly disagree; 7 = strongly agree), with higher scores indicating more perceived control.

Permission Clarity (Clarity). Users' understanding of why an app requests a permission strongly influences their decisions [24, 8, 100]. To assess this, we developed a three-item clarity scale ($\alpha = 0.91$), administered before (Clarity Pre) and after app interaction (Clarity Post). Example items are *"It is clear to me why this app needs access to my {permission protected resource}"* and *"I have no idea why this app wants access to my {permission protected resource}"* (reverse coded). Items are rated on a 7-point scale (1 = strongly disagree; 7 = strongly agree), with higher scores indicating greater clarity.

Additional measurements. Additionally, we recorded the following measures: (1) **Permission Decision:** Participants indicated whether they would grant access with the item *"Based on your interaction with this app, would you grant this app access to your {permission protected resource}?"* (2) **Perceived Purpose:** Participants reported what they thought was the purpose of the requested permission. Response options were *"for the main functionality of the app"*, *"for some additional feature functionality"*, *"do not know"*, or *"for some other reason."* (3) **Rationale Origin:** When rationales were shown, participants indicated who, in their opinion, provided the rationale. Options were *"the mobile operating system"*, *"the app developer"*, or *"some other entity."* (4) **Rationale Recall:** When rationales were shown, participants were asked to recall their content.

Control variables from previous work. Previous research has identified several situational, app-specific, and user-specific variables that may influence permission decisions. To account for these factors, we included the following control variables: (1) **Permission Purpose (main, visible feature, hidden feature):** The purpose associated with a permission request is a major predictor of permission decisions [24, 170, 100, 96, 97]. Therefore, we classified each request into one of three purpose categories. (2) **Permission Sensitivity:** Prior work shows that permissions perceived as sensitive are more likely to be denied [24, 170, 143]. (3) **Privacy Concerns:** Users with stronger concerns about their privacy may be more cautious in granting permissions [24]. (4) **Prior Privacy Experience:** Previous experiences with privacy issues can shape users' attitudes and, in turn, their permission decisions [174]. Next, we describe how these variables were measured.

Permission Sensitivity: We measured the perceived sensitivity of requested permissions using a three-item scale adapted from prior work [36] to fit the context of permission requests ($\alpha = 0.80$). Participants were instructed as follows: *"When using mobile apps, many people find that there are some resource accesses (permissions) that they are generally comfortable granting, some accesses that they are only comfortable granting under certain conditions, and some accesses are too sensitive that they never or only rarely are comfortable granting. Given the information that this app will request access to your {permission protected resource}. Please indicate to what extent you agree or disagree with the following statements."* Sample items are *"In general, I do not feel comfortable granting access to my {permission protected resource}"* and *"The access to my {permission protected resource} is very sensitive to me."* Items were rated on a 7-point scale (1 = strongly disagree; 7 = strongly agree), with higher scores indicating greater perceived sensitivity.

Privacy Concerns: We measured privacy concerns using a 3-item scale from previous work [105], which was originally developed by Smith et al. [144]. We slightly adapted this scale to measure privacy concerns in apps ($\alpha = 0.85$). Sample items are *"Compared to others, I am more sensitive about the way mobile apps handle my personal information"* and *"To me, it is the most important thing to keep my privacy intact from mobile apps."* Items are rated on a 7-point scale (1 = strongly disagree; 7 = strongly agree), with higher scores indicating higher/more privacy concerns.

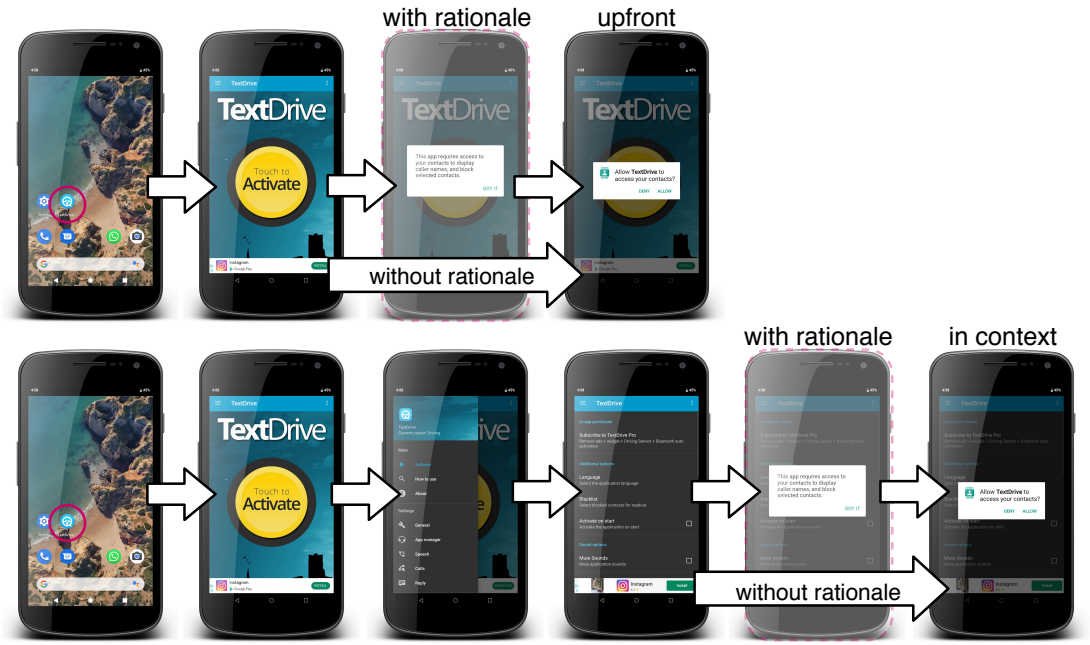


Figure 4.5: Four different variations of the same app depending on timing (upfront vs. in-context) and rationales (present vs. absent).

Prior Privacy Experience: We measured prior privacy experience using a 3-item scale from previous work [144], which was adapted to measure prior privacy experience with apps ($\alpha = 0.80$). Sample items are *"How often have you personally experienced incidents whereby your personal information was used by some mobile app without your authorization?"* and *"How much have you heard or read during the last year about the use and potential misuse of the information collected from mobile apps?"* Items are rated on a 7-point scale (1 = never; 7 = very great extent), with higher scores indicating more exposure to privacy experiences.

Other control variables: Because users might behave differently when they expect and know something, we controlled for the predictability of permission requests, users' familiarity with the app, and user demographics.

4.4.5 App Selection

The user study covered a wide range of apps that requested different permissions for various purposes to rule out possible alternative explanations for our results depending on app-related differences. To achieve that, we selected a set of apps from different categories, each requesting a permission for one of the three permission purposes (main, visible feature, hidden feature). However, we could not rely on the standard Play categories, as apps are organized into superordinate topics, where one topic can contain apps with completely different functionalities (e.g., productivity category contains both barcode scanner and calendar apps). Therefore, we clustered the apps from the empirical

analysis based on their description into 25 clusters using the Latent Dirichlet Allocation (LDA) topic modelling technique and randomly selected 10 clusters for the user study. We then manually choose three apps per cluster, each requesting a permission for one of the three permission purposes. Based on our empirical analysis, we limited the study to the six most commonly found permissions (`MICROPHONE`, `CONTACTS`, `PHONE`, `CAMERA`, `LOCATION`, and `STORAGE`). We excluded any app that required login (e.g., banking and dating apps), since we did not analyze those in our empirical analysis. A list of the apps used in this study and their categories are shown in Appendix 8.4.

In total, we used 30 apps, of which we captured screenshots of the states that led to the permission request. We used these screenshots to create interactive mockup apps that worked similar to real-world apps. Each app was then modified to request a permission for each of the four possible combinations of timing (upfront vs. in-context) and rationales (present vs. absent), resulting in a total of 120 app variations. Figure 4.5 shows such an app with the four different versions.

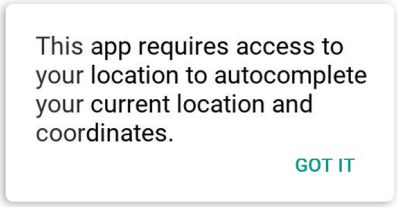


Figure 4.6: Example rationale on Android, created for the user study.

4.4.6 Rationale Selection

To investigate the effect of presence/absence of rationales as they are intended to be [66, 16], we decided to only use rationales with additional information. We also focused on one standardized rationale design to ensure comparability of the results, which was informed by the empirical analysis.

Our study apps showed participants one permission request preceded by a rationale (depending on the experiment version). For that, we chose the dialog design because it has the highest priority in conveying information to the user [107], and because the alternatives are often used for different purposes (e.g., fullscreen views explain multiple permissions and snackbars are displayed after a permission request).

As for the wording of rationales, guidelines of Google and Apple recommend that rationales should use sentence case, be short, clear, accurate, and polite so people do not feel pressured [66, 108, 16]. From the empirical analysis, we extracted rationales that followed these guidelines (e.g., *"Access to camera is required to make new photos"*, *"This app needs your permission to store images to your device,"* and *"This application requires the manage phone call permission to be approved in order to use the favorite store functionality"*) and derived a general sentence structure to use in our study: *This app requires access to your {permission} to {list of purposes}*. A sample rationale used in this study is found in Figure 4.6

Table 4.1: The final multilevel models.

	Decision *Odds Ratio* (*std.* β)	DES Inform β (*std.* β)	DES Satis β (*std.* β)	DES Control β (*std.* β)	Clarity Post β (*std.* β)
User Level					
(Intercept)	2.92 (1.06)**	3.90 (-0.54)***	6.17 (0.24)***	5.34 (0.07)***	4.53 (-0.21)***
Privacy Concerns	0.64 (-0.57)***	-0.02 (-0.02)	0.06 (0.07)	0.00 (0.00)	-0.01 (-0.01)
Prior Experience	1.91 (1.00)***	-0.03 (-0.03)	-0.25 (-0.35)***	-0.29 (-0.32)***	-0.09 (-0.07)***
Request Level					
Purpose (ref: main)					
visible feature	1.35 (0.3)	0.24 (0.18)*	0.03 (0.03)	0.18 (0.13)	0.14 (0.07)
hidden feature	0.35 (-1.05)*	-0.05 (-0.04)	0.07 (0.06)	-0.05 (-0.04)	-0.48 (-0.23)**
Clarity Pre	2.06 (1.53)***	0.18 (0.28)***	0.09 (0.18)***	0.07 (0.12)***	0.59 (0.61)***
Perm. Sensitivity	0.53 (-0.99)***	-0.01 (-0.01)	0.00 (0.01)	-0.01 (-0.01)	-0.03 (-0.02)
Decision Grant	–	0.44 (0.32)***	-0.57 (-0.53) ***	-0.30 (-0.22)***	–
In Context	1.48 (0.39)*	0.30 (0.22)***	0.08 (0.08)	0.06 (0.04)	0.36 (0.18)***
With Rationale	2.73 (1.00)***	0.66 (0.48)***	0.18 (0.17)***	0.07 (0.05)	0.93 (0.46)***
Interaction Tim. & Rat.	–	-0.37 (-0.27)***	-0.20 (-0.19)**	–	-0.37 (-0.18)**
Marginal R^2	0.483	0.211	0.198	0.136	0.504
Conditional R^2	0.765	0.476	0.549	0.679	0.562

Three-level regression model for each outcome variable. The coefficients for Decision are shown as odds ratios, where values <1 indicate a lower likelihood to grant permissions and values >1 indicate a higher likelihood. *std.* β = *standardized* β. $^{*}p < .05, ^{**}p < .001, ^{***}p < .0001$. Decision coding: 0 = *deny*, 1 = *allow*. $N_{User} = 473$, $N_{App} = 30$, $N_{Request} = 1824$. Note that $Level_{App}$ is not shown because the final models do not contain variables from that level.

The extracted sentence structure then had to be filled with meaningful permission purposes for each user study app. For that, we manually ran each app, checked the app's source code, description, and rationale (if available). Then, we manually selected reasonable purposes from a list of most common permission purposes that we extracted from our empirical analysis and related work [102] using Part-of-Speech tagging (POS tagging) [118]. Examples of purposes include: find bus stops nearby, block harassing calls, and use speech translation.

4.4.7 Ethical Considerations

The study design and protocol were reviewed and approved by the Ethics Review Board of our institution. We followed the guidelines for academic requesters outlined by MTurk workers [70]. All server-side software (i.e., Limesurvey Community Edition software) was self-hosted on a maintained and hardened server to which only the researchers involved in this study have access. At the beginning of the study, there was an informed consent procedure, which provided participants with details about the purpose of the study and the type of data being collected. We also informed participants about the option to withdraw from the study at any time.

4.5 Results

We used multilevel regression analysis to evaluate the effects of timing and presence/absence of rationales on users' permission decisions (Decision), the evaluation of their

decisions (DES: DES Inform, DES Satis, DES Control), and the perceived clarity of the permission purpose (Clarity Post). All analyses were performed with R 4.0.2 [133] and the package LME4 [19]. As a data preparation step, we calculated mean scores for measurements with multiple items. We also centered all User Level and Request Level variables by their total mean (grand mean centering) to facilitate interpretation of regression models.

Correlation analysis revealed strong relationships among participants' education, computer science background, familiarity with the app, predictability of the requested permission, and the type of permission requested. We also observed a strong positive correlation between perceived permission sensitivity and participants' privacy concerns: users who value privacy tend to perceive permissions as more sensitive [24]. In addition, we found a significant negative correlation between pre-interaction clarity and the purpose of the permission. This aligns with expectations, as permissions tied to core functionality or visible features are generally easier for users to understand than those linked to hidden features.

4.5.1 Model Construction

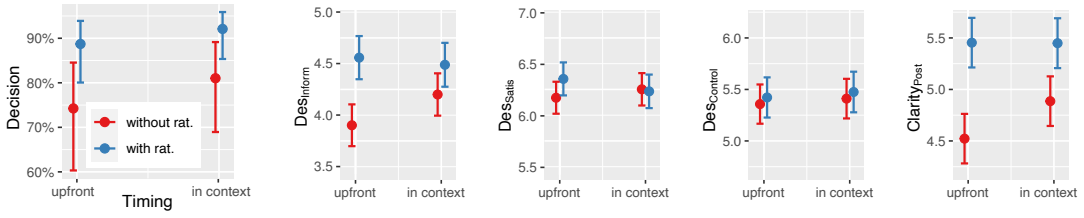
We used a linear multilevel model for DES Inform, DES Satis, DES Control, and Clarity Post, whereas Decision (binary) was modeled using a generalized linear multilevel model. The comparison between a simple and a multilevel regression model showed that multilevel models explain our data significantly better (see Appendix 8.3). To prevent over-parameterization of the models, we built and tested them in a step-by-step approach, following recommendations in the literature [79] in each step. All models were calculated using maximum likelihood estimation to ensure their comparability. Next, we explain the model building process, which was held constant for all outcome variables.

We built our models in four steps. First, starting from a simple regression, we created a random-intercept model by including app and user as random effects. Second, we added all variables identified in prior work: Clarity Pre [24, 8, 100], Privacy Concerns [24], Prior Experience [174], Purpose [24, 170, 100, 96, 97], and Permission Sensitivity [24, 170, 143]. We also included participants' decisions (Decision) as a control variable, since the outcome (grant vs. deny) influences how comfortable users feel with their choice [24]. Third, we introduced our variables of interest: Timing (upfront vs. in-context) and Rationales (present vs. absent). Finally, in the fourth step, we added the interaction between Timing and Rationales when doing so improved model fit. Details of the model-building process are provided in Appendix 8.3.

4.5.2 Final Models

The final models were recalculated using Restricted Maximum Likelihood Estimation, which leads to a more conservative and less error-prone estimation of the parameters [79]. Table 4.1 shows the final model for each outcome variable.

We followed suggestions of literature [3] to identify and handle outliers. We checked for multi-construct outliers on the aggregated App Level and found no conspicuous data points. Then, we checked for multi-construct outliers on the User Level and found 3 participants with conspicuous Mahalanobis distances. We also found 6 outliers on the



(a) Decision **(b)** DES Inform **(c)** DES Satis **(d)** DES Control **(e)** Clarity Post

Figure 4.7: Effects of timing and rationales on each outcome variable. Means were predicted holding all other variables constant at the reference/average level. Error bars represent 95% confidence intervals of the predicted means.[1]

Request Level. Since the removal of outliers did not change the model fits, significance levels, and conclusions, we opted to keep them in the analysis [3]. Additionally, we checked the final models for multicollinearity and found no such case ($VIF < 2$).

Effect of users' individuality. The final models were able to explain 47.6%–76.5% of the total variance in the outcome variables (Conditional R^2), whereby it is worth to note that **a large proportion of this variance is explained by the individual differences between users**. For example, in the final Decision model, intraclass correlation for the User Level was $ICC = 0.490$, which means that 49% of the empirical variance of permission decisions can solely be explained by individual differences between users. The same applies for the DES: DES Inform ($ICC_{\text{User}} = 0.321$), DES Satis ($ICC_{\text{User}} = 0.432$), and DES Control ($ICC_{\text{User}} = 0.625$). In contrast, differences between users in the Clarity Post model only explained 7.8% of the empirical variance, which is due to the fact that we controlled for Clarity Pre in the same model.

4.5.3 Effect of Timing and Rationales

Permission decision. Participants' permission decision was explained best (76.5% of the empirical variance) by a model including the two main variables of interest but not their interaction (*Model Step* 3, $AIC = 1449.35$, $LogLik = -713.68$). We found that both **timing and rationales had a positive effect on grant rates**. When permissions were requested in context, grant rates increased by 48% (*odds ratio* = 1.48, *standardized* $\beta = 0.39$, $p = 0.017$). Additionally, it was 173% more likely that participants granted permissions when rationales were provided compared to permission requests without rationales (*odds ratio* = 2.73, *std.* $\beta = 1.00$, $p < 0.001$). Overall, if permissions were requested upfront and without rationales, they were granted in only 74% of the cases, while they were granted in 92% of the cases if they were requested in context and with rationales (see Figure 4.7a for an overview of the predicted probabilities of granting permissions).

Informed Decision. Participants' perception of having made an informed decision was explained best (47.6% of the empirical variance) by a model including the variables

of interest and their interaction ($Model\ Step\ 4$, $AIC = 5633.44$, $LogLik = -2802.72$). The model shows a significant interaction of timing and rationales ($\beta = -0.37$, $std.\ \beta = -0.27$, $p < 0.001$). Overall, **rationales had a positive effect on whether participants' decision was informed; however, this effect was stronger when rationales were shown upfront** instead of in context. Furthermore, timing was only significant when no rationales were present. This means that **without rationales, requesting permissions in context increases informed decision**, as is depicted in Figure 4.7b.

Decision Satisfaction. Participants' satisfaction with their decision was explained best (54.9% of the empirical variance) by a model including the two main variables of interest as well as their interaction ($Model\,Step\,4$, $AIC = 4695.43$, $LogLik = -2333.72$). The results show a significant interaction of timing and rationales ($\beta = -0.20$, $std.\ \beta = -0.19$, $p = 0.003$). On one hand, **when permissions were requested upfront, rationales had a positive effect on decision satisfaction, but when requested in context, rationales had no significant effect.** On the other hand, **timing had no effect on satisfaction** (see Figure 4.7c).

Decision Control. Participants' perceived control over their permission decision was explained best (67.9% of the empirical variance) by a model that included the two variables of interest but without their interaction ($Model\,Step\,3$, $AIC = 5243.57$, $LogLik = -2608.78$). The results show **no significant effect of timing and rationales on decision control**, as shown in Figure 4.7d.

Permission Clarity. Participants' perceived clarity of the permission purpose was explained best (56.2% of the empirical variance) by a model including the two main variables of interest as well as their interaction ($Model\,Step4$, $AIC = 6418.44$, $LogLik = -3196.22$). After controlling for the initial clarity of permission requests, we found a significant interaction of timing and rationales ($\beta = -0.37$, $std.\ \beta = -0.18$, $p = 0.003$). On one hand, the effect of timing was only significant without rationales, meaning that **post clarity increased when permissions were requested in context without rationales.** On the other hand, **rationales significantly increased permission clarity** for both upfront and in-context permission requests; however, this effect is stronger for upfront requests, as shown in Figure 4.7e.

4.5.4 Effect of Other Variables

Privacy Concerns. Participants' privacy concerns had a negative effect on the likelihood to grant permissions ($odds\,ratio = 0.64$, $std.\ \beta = -0.57$, $p < 0.001$), but not on the other outcome variables. In other words, participants with higher privacy concerns are less likely to grant permissions than those with lower concerns.

Prior Privacy Experience. The data revealed that the more participants dealt with privacy related experiences in the past, the more likely they were to grant permissions ($odds\,ratio = 1.91$, $std.\ \beta = 1.00$, $p < 0.001$). Whereas for decision satisfaction, decision

control, and clarity of the requested permission, and more privacy-related experiences decreased the score of these scales. Only for informed decision, we could not find an effect of prior privacy experience.

Permission Clarity. Participants' initial clarity of the permission purpose had a significant effect on all outcome variables. Having an initial understanding of the permission purpose increased the odds of granting permissions by 106% ($odds\,ratio = 2.06$, $std.\,\beta = 1.53$, $p < 0.001$). Also, for all three DES, a positive effect of initial clarity was found. Furthermore, the clearer the permission request was before interacting with the app, the clearer it was afterwards ($\beta = 0.59$, $std.\,\beta = 0.61$, $p < 0.001$).

Permission Sensitivity. There was a negative effect of permission sensitivity on decision ($odds\,ratio = 0.53$, $std.\,\beta = -0.99$, $p < 0.001$). Meaning that permissions perceived as sensitive are less likely to be granted.

Permission decision as a control variable. As for the effect of permission decision, we found that granting a permission increased the perception that the decision was informed, while it decreased decision satisfaction and the perception of being in control.

Effect of other control variables. To rule out potential alternative explanations for our results, we built additional models to examine whether there were any changes in the outcomes due to the ordering of permission requests, having interacted with the app before, and the predictability of permissions. None of these control variables significantly changed the effect of timing and rationales on the outcome variables. Also, we did not find a significant effect of gender or age. Neither did participants' education, having a computer science background, nor participants' mobile OS explain any additional variance in our data. Additionally, we built the DES models with and without Decision as a control variable and found no significant difference in the effect of timing and rationales.

4.5.5 Rationale Recall

To further rule out potential alternative explanations for our results, we built the models again for attentive participants only. For that, two researchers analyzed and independently coded the free text answers of participants' ability to recall the content of the rationale messages. The analysis showed almost perfect inter-rater agreement between the two coders ($Cohen's\,\kappa = 0.87$) and all differences were resolved in agreement. Four themes emerged in the coding process: (1) Participants correctly recalled all or parts of the rationale message (correct), (2) they did not recall the content of the rationale and provided unrelated responses (unrelated), (3) they admit to have forgotten the content of the rationale (forgotten), or (4) they claim to have not seen the rationale dialog (unseen). From all rationale recall answers ($N = 899$), 49% were coded as correct,

[1] Due to our within-subject design and the resulting paired data, the confidence intervals from Figure 4.7 cannot be interpreted as an indicator of the statistical significance of the main/interaction effects [34].

45% as unrelated, 5% as forgotten, and 1% as unseen. These percentages reflect the user's general inattention to security and privacy-related information [138, 6, 26] that would have also occurred if participants interacted with the apps on their real phones. Each model was built again for attentive participants who recalled the content of at least one of the rationales. We found that the effect on timing and rationales was consistent and did not change. The only difference was that rationales had a significant effect on DES Control. In order to stay on the conservative side, we only considered the results of the main analysis.

4.5.6 Rationale Origin

Participants were asked once about the rationale origin for each app that displayed a rationale. However, since each participant interacted with two apps with rationales, we only considered the last response given. We found that 57% (270) of the participants correctly identified the app developer as the provider of the rationale, while 37% (175) thought that it came from the operating system. We checked whether the operating system of the participant's mobile phone was one of the reasons for this misunderstanding, which was not the case. The remaining 26 participants said that they do not know who provided rationales, and 2 gave unrelated answers.

4.5.7 Permission Purpose

We found that asking participants about the purpose of permissions did not yield useful insights, as the responses reflected participants' subjective perception of permission purposes. Therefore, we do not report on the results.

4.6 Discussion

Our study is the first to explore the effect of timing and rationales and their interplay on users' runtime permission decisions and the evaluation of their decisions. We found that timing and rationales matter even after accounting for user and app-level differences identified in previous work. In addition, we showed that timing and rationales should not be evaluated in isolation because both might influence one another. We also found that a large proportion of the variance in the outcome variables can be explained by the individual differences between users.

Effect of timing. Requesting permissions in context primarily benefits developers, as such an approach increases grant rates. Whereas requesting permissions in-context only has a positive effect on users' perception of their decisions without rationales.

Effect of rationales. Requesting permissions with rationales benefits both developers and users, as such an approach increases grant rates, helps users in making informed decisions by increasing their understanding of the permission purpose, and positively affects decision satisfaction. Whereby, the benefits of rationales are greatest for upfront requests, when users may lack contextual data for decision making.

Alternative to Google's guidelines. Google's guidelines recommend four strategies to help developers minimize deny rates [106]. They suggest requesting app-critical permissions upfront and function-specific permissions in context, and providing rationales for permissions that may be unclear. While these suggestions appear straightforward, our study and previous work [143] show that permission clarity is a subjective measure. Consequently, it is unrealistic to expect developers to accurately determine which permission requests might be unclear to end users and therefore require a rationale. Moreover, our results indicate that some permission request strategies are, on average, less effective than others. For instance, requesting permissions upfront without a rationale leads to the lowest grant rates and the least positive perception of permission decisions. In contrast, including rationales (whether upfront or in context) benefits both developers and users: developers see higher grant rates, while users make more informed, understandable, and satisfying permission decisions.

Based on these findings, we propose an adjustment to Google's guidelines. Instead of offering four permission request strategies, we recommend limiting developers to two: request permissions either upfront with a rationale or in context with a rationale. In contrast to the current guidelines, we suggest that rationales should *always* be provided, while retaining the existing recommendation to request app-critical permissions upfront and function-specific permissions in context. This simplification is expected to maintain high grant rates while simultaneously increasing users' comfort with their runtime permission decisions.

Individually tailored system support. Google's guidelines put the burden on developers to decide when to request permissions (timing) and whether to provide further explanations (rationales). Even with our improvements, developers' still have to time permission requests for all users. Additionally, our results showed that users differ in their decisions and the way they make those decisions, led by their own values and preferences. So, instead of a strategy that attempts to fit all users with the burden on developers, our intuitive deduction is to provide a solution to support users' individuality.

One concrete suggestion is to enable the operating system to customize permission requests on a per-user basis. Through system settings, users could specify when they want to be asked for permissions and whether they prefer to see rationales. Developers, in turn, would only need to follow a standardized pattern: labeling the in-context positions for permissions and providing a list of rationales (similar to iOS [15]). A key advantage of this consistent approach is that users would not be surprised or annoyed by unexpected requests, as they would know when to anticipate them. Since the permission request mechanisms on Android and iOS are largely similar, this solution could be implemented on both platforms. However, any such design change to the operating system would also need safeguards against malicious developers providing misleading in-context timings or rationales, a challenge beyond the scope of this work. The concrete design and evaluation of such systems remain directions for future research.

Rationale origin misconception. While the majority of participants identified the developer as the author of rationale messages, a large number still thought that the rationales were created by the operating system (37%). This could be a side-effect

of using standardized rationales for the apps in our user study. However, rationale messages in iOS are already integrated in the standard permission dialog [15]. Therefore, we recommend adding an indicator that the rationale is provided by the app developer. This could be a short message proceeding the rationale. For example: "{App name} says: {Rationale message of the app developer}." However, this solution is only applicable when the rationale is standardized by the operating system, as in iOS. Whereas in Android, currently only the app developer is able to highlight the origin of the rationale (e.g., through custom themes and wording).

Generalizability of our findings. When interacting with modern technology, users are often confronted with security and privacy-relevant decisions. Such decisions must be informed while being consistent with users' individual values and preferences. To offer users more transparency, previous research focused on providing comprehensive privacy policies (e.g., in the form of "privacy nutrition facts" [85]) and effective browser security warning messages [153, 6, 39].

Consistent with these findings, we observed that users made better-informed and more satisfying permission decisions when provided with transparency, most notably through rationales, and to a lesser extent through appropriately timed requests. Our results align with prior work in other areas of the mobile domain, such as presenting security-related behavior in app descriptions [175], explaining permission usage based on code [132], and supporting users in the app-selection process [73, 95, 87]. All of these studies underscore the critical role of transparency in users' decision-making. This importance may also help explain recent initiatives by major mobile operating systems to enhance app privacy and security transparency, for example through the introduction of "privacy labels" in iOS and the upcoming safety section in Google Play [50].

In line with these efforts to aid users in their decisions, we recommend that rationales should always be provided by developers. However, future research is needed to optimize how frequently they are displayed to the user, e.g., leveraging machine learning to learn individual preferences [171, 119]. For example, depending on users' individual preferences, a user who always denies a certain permission or always denies permission for certain app types may not need additional rationales in these situations. We believe that our findings on rationales are also applicable to other security and privacy critical decisions. While how rationales should look like is system dependent, they all need to strike a balance between adequately informing and overwhelming users. Since our results show that just the presence of rationales is beneficial, future work could study the magnitude of this effect depending on different rationale designs and contents.

4.7 Threats to Validity

As with any empirical study, there are aspects of our study design that might limit the generalizability of results. First, our data was collected in a highly standardized, somewhat artificial situation. Therefore, it might be fair to question whether our results fully reflect the behavior of users on real apps. However, only such experimental research methods that provide conscious control of all aspects of a situation (high internal validity), allow the direct inference of causal relationships [145]. To address potential negative

effects of this design decision, we followed best practice recommendations for this kind of experimental studies [2]. For example, our participants were given a consistent storyline and clear goals they should reach with their apps as well as interactive mockup apps. These measures ensure a high level of immersion for participants, which, as prior work has shown, leads to the highest possible generalizability of the study results [2, 172, 80].

Second, our research topic – permission requests – was obvious to our participants at several points in our study, which may have primed their behavior in a certain way. For example, we asked participants about a permission prior app interaction (making them aware that the app will request this permission). This was necessary, as some variables (i.e., permission sensitivity/predictability/clarity) could only be accurately measured before users interacted with the app. However, from the users' perspective this is very similar to checking permissions in the app store before installing the app. Another priming could have resulted from the fact that each participant went through the main part of the study for several apps. We mitigated potential carryover- and order-effects arising from this within-study design by randomizing the order of the permission request types (upfront vs. in-context, with vs. without rationale) and checking that the order did not affect our results.

4.8 Conclusion

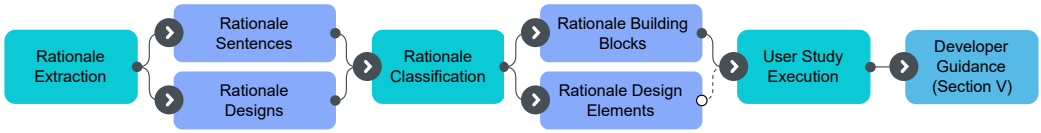
In this work, we showed that timing of permission requests and presence/absence of rationales have an effect on users' permission decisions and the evaluation of their decisions. We found that the effect of timing and rationales depend on one another and should not be evaluated on their own. Based on the results, we suggest that the current Google guidelines should be refined to better aid users in their decision-making process. Further, we highlight that permission decisions mainly depend on the individuality of users, suggesting that there is no one-fits-all permission request strategy. As a conclusion, current mobile platforms could benefit from a customized solution on a per-user basis, in which users can define when permissions should be requested and whether rationales should be given.

5
Rationale Phrasing

The Power of Words: A Comprehensive Analysis of Rationales and Their Effects on Users' Permission Decisions

5.1 Motivation

Imagine you open an app, and one of the following messages pops up: *"We would like access to your camera for the app to work correctly,"* or *"Please allow access to your camera. Without this permission, the app cannot scan your documents. We do not collect or transfer any personal data outside your phone."* Which permission request would you rather approve? Which would leave you feeling better informed, more satisfied, and more in control of your decision? In a world where app developers have the freedom but also the responsibility to clarify their permission requests to users, it is crucial that the wording and presentation of these explanations align with users' expectations.



(a) Section 5.3: Investigating Rationales **(b)** Section 5.4: User Study

Figure 5.1: Methodology of our exploration of rationales and their effect on users' permission decisions.

5.2 Contribution

In this work, we explore *"rationales"*—the explanations behind runtime permission requests—by analyzing their designs, wording, and how different phrasings affect users' permission decisions. **Our first goal is to understand the current state of rationales in Android apps** (see Figure 5.1a). Despite numerous guidelines for developers on effective permission communication [117, 11, 66], the practical implementation and interpretation of these recommendations in the design and phrasing of real-world rationales remains uncertain. By manually analyzing 720 sentences and 428 screenshots of rationales collected from the top apps on Google Play, we show how app developers convey their permission needs in practice. This investigation establishes a baseline for evaluating future innovations and changes in this domain, potentially guiding the improvement of existing best practices for the benefit of both developers and users.

Our exploration revealed the diversity of rationales with various patterns. We found that app developers phrase their rationales from different perspectives. Rationales can comprise one or multiple phrases. They can be specific or vague, positive or negative, and may include additional information as optional clauses. Regarding design, we observed diverse layouts with unique combinations of buttons, icons, and titles. Additionally, we observed patterns that occurred more frequently among developers, such as using a dialog to present a concise rationale, typically composed of a single phrase.

Finding a variety of different ways how app developers can convey permission needs to users naturally leads to **the second question of whether different rationale phrasings impact users' decision-making process** (see Figure 5.1b). Previous research in other fields has shown that linguistic variations can influence a variety of user decisions [58, 5, 37, 159, 90, 141]. Therefore, we break down rationales into their

fundamental building blocks and examine how different combinations of these building blocks affect users' permission decisions and their perception of those decisions.

In an online user study with 960 participants, we gain insights into how users perceive and respond to rationales. We demonstrate that rationale phrasings alone significantly impact user choices regarding permissions and their assessment of these choices. When comparing the two rationales at the outset of this introduction, our study reveals that the second phrasing leads to a higher likelihood of granting permission, provides users with more informed decision-making, increases user satisfaction, and enhances the perception of being in control.

We observed that the natural variability in rationales is inherent. However, specific phrasings within rationales are essential for app developers to prioritize to improve the user's decision-making process. Finally, we compare our findings with available guidelines to create actionable recommendations for app developers, aiming to make rationales a usable security instrument for users.

5.3 Investigating Rationale Differences

While numerous recommendations exist for developers about how to communicate permission needs to users [117, 11, 66], the actual interpretation and implementation of these recommendations in rationales remain unclear. The following exploratory analysis aims to unveil how permission needs are communicated in practice. To this end, we first crawled the most relevant apps from Google Play. Next, we extracted rationales and manually labeled them in a bottom-up coding process. Finally, we extracted dimensions along which developers' implemented rationales typically differ.

Between 08/2021 and 04/2022, we continuously crawled the top 50 apps in every category on Google Play. This effort produced a dataset containing 11,500 unique APKs, considering solely the most recent app versions. For our exploratory analysis, we narrowed this dataset to apps that requested at least one runtime, i.e., dangerous protection level, permission in their manifest file. This yielded a final selection of 9,489 APKs, set for in-depth exploration.

5.3.1 Extraction and Classification of Rationales

Our next step was to extract rationales. Rationales can come in various forms and may not be immediately recognizable within the app's UI elements. However, we saw this as a valuable starting point for locating rationales because Google recommends that developers specify all text elements in the strings.xml files of Android apps [63]. Given the challenge of identifying rationales due to their scarcity, we developed an initial classifier using SpaCy [44]. This classifier was trained on a labeled dataset from our prior work [P1], which included 450 rationales and 250 non-rationales. We employed this classifier to filter out non-rationales from all sentences extracted from the string.xml files, resulting in 55,000 unique sentences that had the potential to be rationales.

During the initial classifier assessment, we manually checked 3,945 of these sentences. Once labeled, cleaned, and cleared of duplicates, these sentences comprised 801 rationales and 892 non-rationales set aside for threshold optimization and evaluation.

The remaining sentences underwent annotation using Prodigy [43], a tool that employs active learning, selectively prompting users to annotate sentences that the classifier struggles with. In total, 1,500 sentences were annotated—777 as rationales and 723 as non-rationales—forming our balanced training dataset. Our final classifier achieved a precision of 0.99, a recall of 0.84, and an F-score of 0.91.

Applying this classifier to the strings.xml files, we found 35,737 unique rationale sentences across 6,524 apps. Our investigation then broadened to analyze the overall context in which these rationales appeared through dynamic analysis. This involved exploring design aspects and phrases that, while not rationales on their own, carry significance, such as *"See how we protect your privacy, tap here."* or *"We will not collect your personal information."*

To execute dynamic analysis on our dataset, we employed an approach inspired by prior research [30, 29]. Our analysis was conducted simultaneously on four Android emulators (API level 30), with a timeout of 12 minutes per app. We started with decoding the APK and rendering its activities accessible to external entities by setting *"exported=true"* for each activity within the manifest file. We used Apktool [81] to decode and repackage APKs. Then, we launched each activity through adb shell commands and automatically navigated through its interactive components. Permission requests encountered during the launch were intentionally denied and followed by an activity restart. This was essential to activate rationales that emerge specifically after at least one permission denial [59]. Subsequently, we dumped the XML layout of the current activity and used our classifier to analyze all sentences within it for potential rationales. When detected, we captured a screenshot of the respective activity.

Given the complexity of Android activity layouts, potentially including hidden elements, we harnessed UIAutomator [69] to interact with interactive components. Our approach extended to testing swipe gestures, especially relevant when dealing with onboarding processes devoid of clickable *"next"* buttons. Interactions that led to layout changes within the same activity prompted a subsequent rationale check.

We successfully explored 3,818 of the 6,524 apps, encompassing partially analyzed apps due to timeouts (346 APKs). Unfortunately, analysis was unfeasible for 1,486 apps, as their activity initiation necessitated extra parameters ascertainable only through time-consuming static analysis [29]. Apktool could not repack the remaining 1,220 apps.

In total, we collected 2,953 screenshots. We filtered out screenshots depicting cookie notifications, update messages, privacy policies, and other instances erroneously included by our classifier due to resemblances with rationales. While some of these instances did contain rationales, they were often embedded within privacy policies, which were beyond the scope of this study. We also removed duplicates within apps. This curation process resulted in a total of 1,054 unique rationale screenshots from 709 distinct apps.

We analyzed both the content and design of rationales using inductive and axial coding until saturation. Initially, our dataset consisted of 35,737 unique text-based rationales and 1,054 rationale screenshots. Given the substantial number of rationales to code manually, our methodology involved assigning categorical labels (aka. codes) to rationale messages and designs. In this process, two independent researchers created codes for the same batch of rationales, which were then collaboratively discussed until an amalgamated codebook was formed. The final codebook was subsequently used to

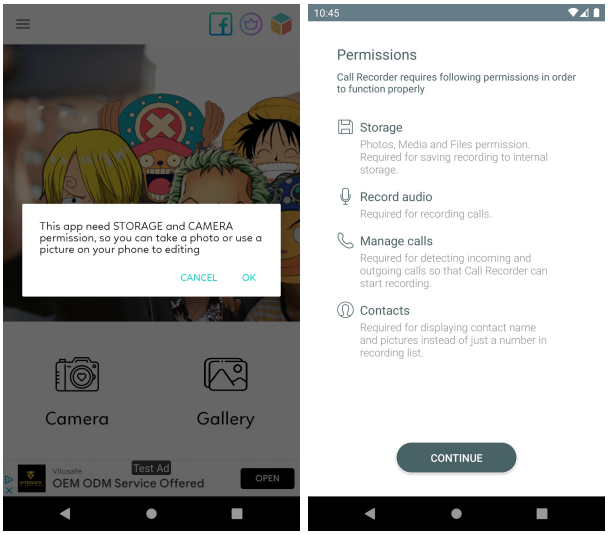


Figure 5.2: (a) Many permissions for one function. (b) One permission per function.

code 720 randomly chosen text rationales and 428 screenshots, creating our sample dataset. The entire coding process took two independent researchers four weeks to complete. The final codebook can be found on the Open Science Framework [41].

Next, we will outline our findings on the diversity of rationale phrasing and design. We will present the occurrence percentages of each label in two datasets: our manually labeled rationale sentences ($N = 720$, marked as $\%^{m}$) and our manually labeled screenshots ($N = 428$, marked as $\%^{s}$).

5.3.2 Rationale Building Blocks

This section focuses on the structure and phrasing of rationale sentences.

Functionality. In our sample, rationales varied between providing specific information about permission use and remaining vague. At times, they failed to reveal the functionality requiring permission. In our manual analysis, when a rationale pertains to a particular feature within the app (e.g., searching for gas stations), we considered it specific ($58\%^{m}$ $69\%^{s}$). Conversely, a functionality that merely indicates that permission is necessary would be labeled unspecific ($12\%^{m}$ $14\%^{s}$). In some cases, no functionality was indicated ($30\%^{m}$ $17\%^{s}$).

Articulation. There are two ways to convey permission-enabled functionality to users. One approach is positive phrasing ($79\%^{m}$ $83\%^{s}$), where granting permission enables the functionality. The other is negative phrasing ($21\%^{m}$ $17\%^{s}$), where lack of permission results in the functionality being unavailable. For instance, positive phrasing would be something like *"Camera permission is needed to use the scanner,"* while its negative counterpart would be *"Unable to use the scanner without the camera permission."*

Permission Type. We observed that rationales typically specify the permission they pertain to (83%^{m}). In our dataset, the most common permissions were location (32%^{m}), storage (29%^{m}), camera (19%^{m}), and microphone (8%^{m}). Rationales can also cover multiple permissions (9%^{m}, 7%^{s}), which fall into two categories: either multiple permissions collectively enable the same functionality, as shown in Figure 5.2a, or each permission individually supports a different functionality, as illustrated in Figure 5.2b.

Perspective. A rationale can be phrased from one or multiple perspectives. It may directly address the user and prompt them to take action, emphasize the app's need for a permission, or highlight how the permission enables a particular functionality.

User Perspective: A rationale framed from the user's perspective encourages them to take action toward granting permission. These prompts can range from simple requests, such as *"please grant permission,"* to more specific guidance, like *"click allow to grant permission"* or *"grant permission on the next screen."* In our dataset, most rationales included a user-perspective phrase (67%^{m}, 59%^{s}).

When permission is blocked, users must take extra steps to grant it. Rationales associated with this situation usually direct users to device settings, e.g., *"please grant permission from settings"* (24%^{m} 18%^{s}). More precise instructions lay out these directions in a step-by-step manner, utilizing commas (, , ,), operators that are used as arrows (>>>), or numbered lists (1. 2. 3.). For example, a direction might read: *"Tap settings > go to app info > permissions, then allow permission."*

We have identified two distinct approaches to how a rationale addresses the user. The first involves employing imperative commands with words like *"grant," "turn on," "allow,"* and *"enable."* This kind of rationale can come across as demanding. However, adding the word *"please"* to the request introduces a more polite tone. We found that around half of the rationales with the user perspective were politely phrased (43%^{m} 40%^{s}). The second approach directly addresses the user using phrases like *"you must," "you will need to,"* and *"you have to."* Furthermore, the users' perspective can also be conveyed in negative sentences like *"you denied permission."*

App Perspective: Another common perspective in rationales is that of the app itself. (30%^{m} 37%^{s}). In this phrasing style, we encountered several variations. The app explicitly expresses its need for permission, often stating *"this app needs permission"* or *"{app_name} needs permission."* Sometimes, specific functionalities of the app need permission, as seen in *"scanning QR codes needs permission."* A more polite approach would be, *"this app would like permission,"*. However, we did not come across this polite variation very often in rationales that included a phrase from the app's perspective (2%^{m} 3%^{s}). The app's perspective is also reflected in negative sentences such as *"the app does not have access to permission."*

Objective Perspective: When the need for permission is communicated passively, such as in the phrases *"permission is needed"* or *"permission is used,"* the rationale takes on an objective tone (27%^{m} 30%^{s}). We also found instances where the request to grant permission is presented in a passive form, as evident in *"permission must be granted."* Additionally, when expressed negatively, examples include *"permission denied"* or *"without this permission, the app cannot function."*

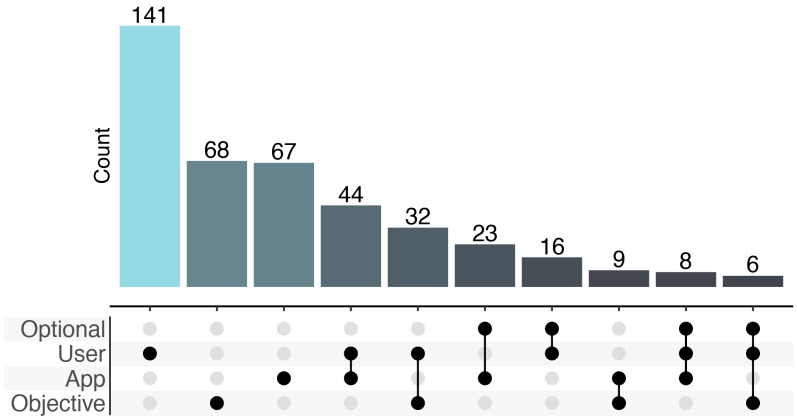


Figure 5.3: Most frequent optional blocks and perspectives (data: screenshots $N = 428$).

User-Centric. A rationale can center around the user by aligning the permission with the user, as in *"access to your location"* ($29\%^{\mathrm{m}}$) or by emphasizing the benefits of granting permission for the user, like *"to search for gas stations near you"* ($18\%^{\mathrm{m}}$ $17\%^{\mathrm{s}}$).

Optional Building Blocks. Apart from the above core component of a rationale, we discovered that a rationale can encompass one or several optional building blocks:

Empower with control (control): One such element involves empowering users to manage their permission choices at any point, such as *"you can change permissions from device settings anytime"* ($1\%^{\mathrm{m}}$ $3\%^{\mathrm{s}}$).

Mitigate (mis)use (guarantee): The rationale can reassure users about the permission's purpose, e.g., *"we will only use your permission for smart tracking,"* or clarify what it will not be used for, like *"we do not track your location"* ($2\%^{\mathrm{m}}$ $6\%^{\mathrm{s}}$).

Offer alternatives: An alternative solution can be integrated into the rationale, giving users an option if they choose not to grant permission. For example, a rationale might state *"alternatively, you can set your location manually"* ($2\%^{\mathrm{m}}$ $3\%^{\mathrm{s}}$).

More information (more): A link to more information or the app's privacy policy can be included ($1\%^{\mathrm{m}}$ $6\%^{\mathrm{s}}$).

Prompt engagement: The rationale can pose a question, prompting users to grant or confirm denial of permission, e.g., *"do you want to allow this permission?"* or *"are you sure you want to deny this permission?"* ($3\%^{\mathrm{m}}$ $6\%^{\mathrm{s}}$).

Rationale Timing. Rationales can be linked to first-time permission requests (type I). They can also belong to previously denied requests (type II), often mentioning that the app lacks permission. Additionally, some rationales are connected to permission requests that have been blocked (type III). This can happen if the user denies permission multiple times within the same app life-cycle or if they have selected the *"never ask again"* option for pre-Android 11 apps. These rationales may include phrases suggesting the user can grant permission from the device settings.

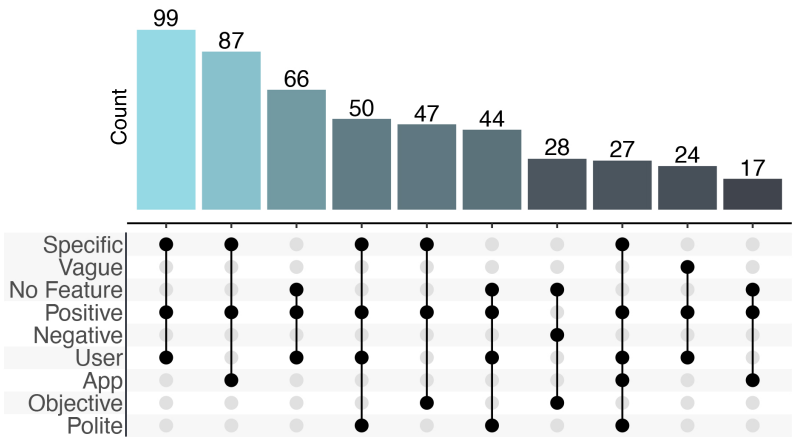


Figure 5.4: Most common rationale building block sets (dataset: sentences $N = 720$).

Multiple Phrases. While rationales are typically short, in some cases, they consist of two or three phrases (34%^{m}, 33%^{s}). When rationales consist of multiple phrases, the most common combination involves a phrase from the user's and one from the app's perspective, as demonstrated in the labeled rationale in Figure 5.5 (cf. Figure 5.3). Additionally, rationales may include optional building blocks.

Figure 5.5: Labeled rationale with two phrases.

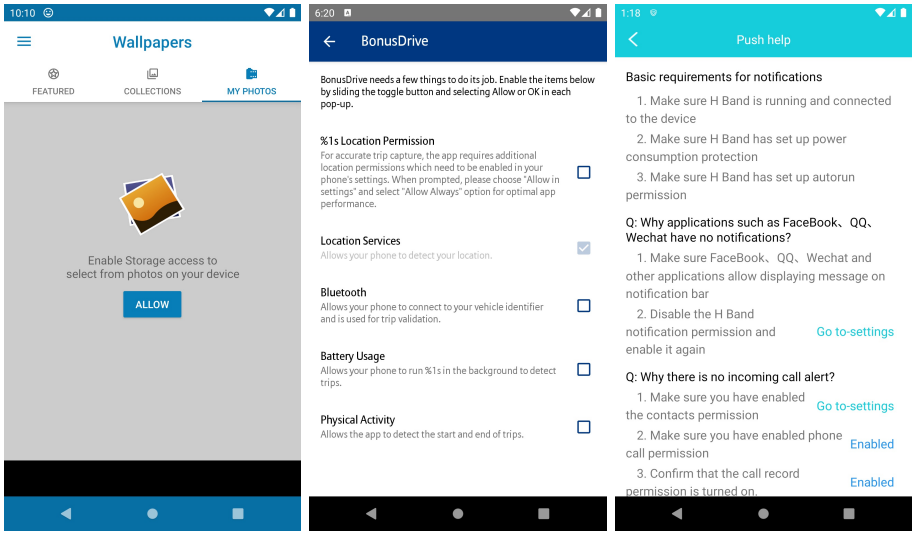
Summarizing our findings on rationale phrasing (Figure 5.4), most rationales consist of a single phrase. Perspectives are mainly user-focused, followed by app- and objective-focused. Many rationales also specify a functionality, indicating that granting permission enables it, a strategy we call positive articulation.

5.3.3 Rationale Design Elements

Next, we will outline our findings concerning the design aspects surrounding a rationale.

Presentation. In exploring 428 rationale screenshots, we discovered that rationales can take on diverse formats, as shown in Figure 5.7. The most prevalent choice is a dialog-style rationale (59%^{s}), followed by fullscreens (23%^{s}), which were sometimes included in the onboarding process, embedded forms (10%^{s}), and banners (8%^{s}).

Dialog. Dialogs typically follow standard Android styling [61], allowing developers to use them without modifications, as seen in Figure 5.7a. However, they can also be customized to resemble fullscreen rationales.



(a) Main **(b)** Settings **(c)** Help

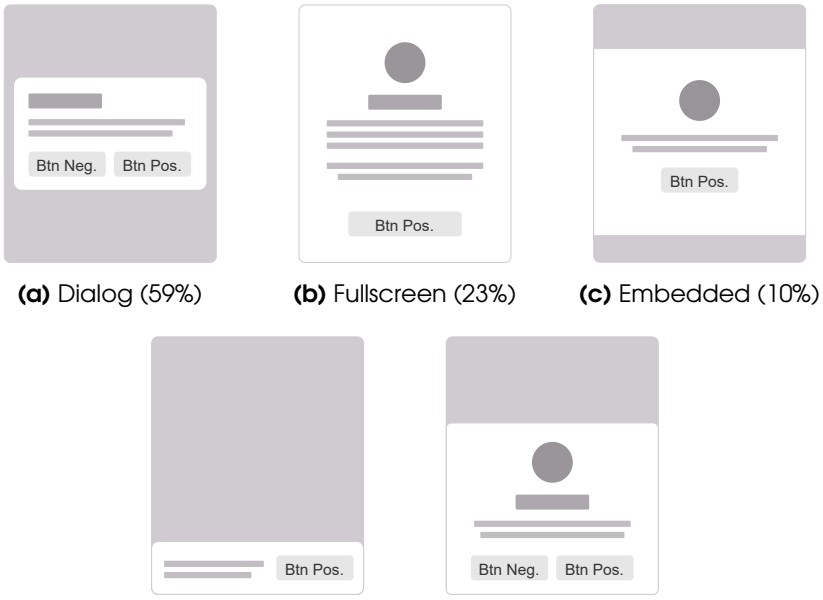
Figure 5.6: Different rationale embedding points in the app.

Fullscreen. Fullscreens lack a default style, with their appearance based on the developer's preferences. Their spacious layout allows for more detailed information (Figure 5.7b). A variation includes integrating the rationale into the app's onboarding process (44%^{s}), typically shown at first launch unless skipped.

Embedded. Embedded rationales are integrated into the app's main screens. They often replace permission-protected content until permission is granted, as shown in Figure 5.7c (cf. Figure 5.6a). Additionally, they can be part of the settings screen (see Figure 5.6b). This rationale stands out due to its interactive nature. It allows users to activate or deactivate permissions using buttons, checkboxes, or toggle switches, visually indicating the permission's status. Alternatively, a rationale can be integrated into a help or troubleshooting screen (see Figure 5.6c).

Banner. There are two forms of Banners. The first form is narrow and includes the rationale message and a button, as in Figure 5.7d. This style is called modeless, meaning it does not interrupt the user's ongoing activity and usually follows the standard Android appearance of banners. Half of the banners followed this form (50%^{s}). The other half took up more space and were modal (50%^{s}), meaning they require user interaction, as in Figure 5.7e.

Buttons. A rationale can feature one or multiple buttons. In cases where only one button is available (54%^{s}), its actions vary. Usually, when this button is labeled with terms like *"ok," "allow," "grant," "enable," "next," "continue,"* or *"proceed,"* it serves a positive function and triggers the display of a permission request. However, this positive button may guide users to the settings screen for rationales related to blocked permissions, with labels like *"settings"* or *"go to settings."* Occasionally, this button



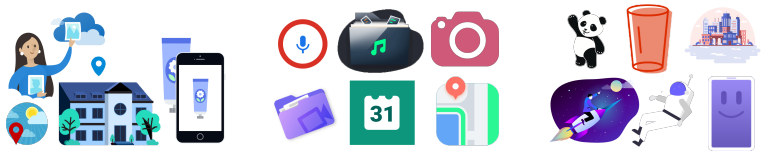
(a) Dialog (59%) **(b)** Fullscreen (23%) **(c)** Embedded (10%)

(d) Banner Modeless (4%) **(e)** Banner Modal (4%)

Figure 5.7: The different rationale presentations. Pos.=Positive, Neg.=Negative.

can also serve a neutral function, dismissing the rationale without further steps. It is often labeled with phrases like *"ok"* or *"got it."*

Additionally, when a negative button is present alongside the positive button (46%^{s}), it is utilized to prevent the permission request from emerging. This button can also manifest as a dismiss *"x"* button located in the upper right corner of the rationale (4%^{s}). The negative button is often labeled with phrases like *"cancel," "deny," "don't allow," "later," "skip,"* or *"close."* In rare cases, an alternative button might replace the negative button, like a button to manually enter the current location (2%^{s}).



(a) Feature-related **(b)** Permission-related **(c)** Unrelated

Figure 5.8: Icon & image types in rationales.

App developers might employ an opinionated design for buttons to motivate users to grant permission. When there are multiple buttons, one approach is to make the positive button stand out more prominently than the negative button (19%^{s}).

Images & Icons. Rationales can include images and icons (33%^{s}). We discovered that these visual elements are often (89%^{s}) directly related to the protected functionality (see Figure 5.8a) or the requested permission (see Figure 5.8b). In other instances (11%^{s}), these images and icons might be the app's logo or for visual appeal (see Figure 5.8c).

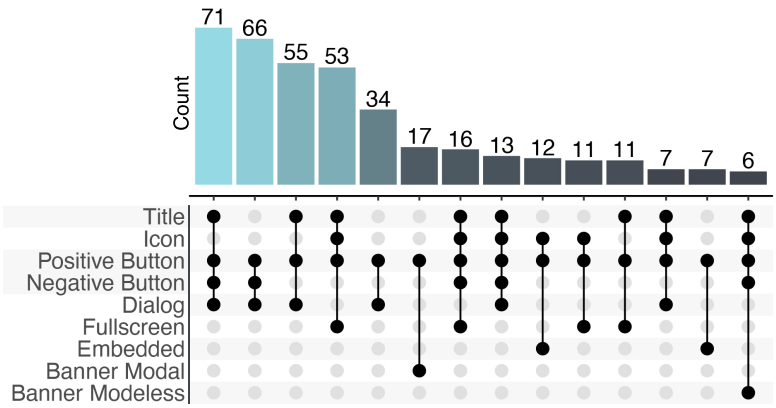


Figure 5.9: Most common design element combinations (dataset: screenshots $N = 428$).

Title. The presence of a title (61%^{s}) depends on the rationale's layout. Generally, embedded rationales, modeless banners, or concise dialogs do not include titles. In most cases, the title aligns with the content of the rationale message, using phrases like *"permission required"* or *"grant permission"* (54%^{s}). Alternatively, a more neutral wording could be *"permission request"* or *"location permission"* (26%^{s}). Occasionally, the title serves as a welcome message to the app or states the app's name (8%^{s}). Titles can also serve as attention-grabbers; we observed that some titles incorporated an exclamation mark icon, prompting users to take action, as seen in examples like *"Warning!," "Attention!,"* and *"Action Required!"* (12%^{s}).

Summarizing the rationale designs in Figure 5.9, we observe distinct patterns across different layout types. Dialogs are the most common, typically including a positive button, a negative button, and often a title. Fullscreens, taking advantage of additional space, can include extra information, visual elements, a title, and usually a positive button only. Embedded rationales often incorporate an icon or image along with a positive button. While modeless banners contain only a positive button, modal banners can include a title, a visual element, and both positive and negative buttons.

Finding: Our investigation of rationales in Android apps showed considerable variation in how developers implement rationales in terms of phrasing and design. Nevertheless, we also identified some common trends that developers followed more frequently. Many developers preferred using dialogs to present rationales. Furthermore, we observed that rationales tend to be concise, typically composed of a single phrase from one perspective—user, app, or objective.

5.4 User Study

Our exploratory analysis in revealed a variety of patterns in how app developers present rationales. Building on this insight, we conduct a user study to gather direct user feedback on various rationale phrasings. Our study evaluates users' permission choices, their understanding of these decisions, satisfaction levels, and perceived control.

Ultimately, we aim to establish practical guidelines that aid app developers in effectively communicating permission requirements to enhance the efficacy of rationales and improve overall user experience.

5.4.1 Study Design

We structured the study as an online experiment using a repeated measures (within-subject) design. This approach was chosen to minimize errors related to individual differences, which are often misrepresented in between-subjects designs in judgment-related studies [21]. Each participant interacted with rationales for the four most common permissions—location, storage, camera, and microphone—as discussed in Section 5.3.2. These rationales were constructed by combining different rationale building blocks derived from our exploratory analysis, detailed below. To account for potential similarities among observations from the same user, we implemented a multilevel design, illustrated in Figure 5.10. Additionally, we randomized the sequence of rationales and permissions to mitigate order effects.

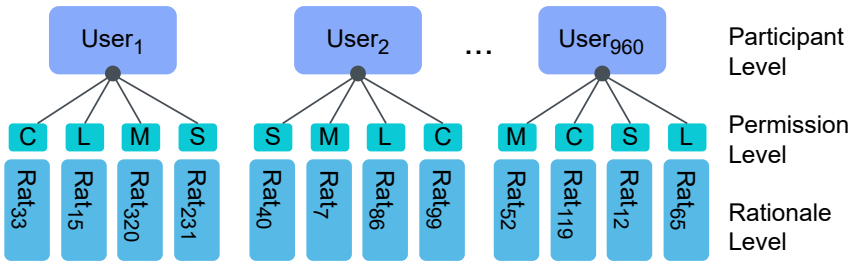


Figure 5.10: Hierarchical structure of the user study. Permissions: C=Camera, L=Location, M=Microphone, S=Storage.

We maintained a consistent rationale design throughout our user study. This approach acknowledges that testing all conceivable designs and formulations in a single user study would be economically and statistically infeasible. Therefore, our strategy involves identifying the most effective phrasing for rationales, which can then be tested across diverse designs in future research. This strategy mirrors recent research that separately examines phrasing and design patterns [58]. Additionally, the appearance of rationales is often tied to a specific application or platform and cannot be uniformly governed. For the user study, we used dialogs because they are most commonly used for rationales, look similar across different apps, and are suitable for sharing important information that needs quick attention, like granting or denying permission.

Rationales can accompany permissions requested upfront (e.g., at app launch) or in context (e.g., button click). Previous research has shown that rationales overpower the effect of timing [P1], prompting our focus on upfront rationales. Additionally, we focused on type I rationales identified in our exploratory analysis, which are presented before permission is requested and visible to all users. In contrast, type II and III rationales are shown only to users who deny permission, potentially multiple times.

We would like to emphasize that our study was preregistered to enhance transparency and credibility. Preregistration enabled us to define our objectives, sample size consider-

Table 5.1: Rationale building blocks for the user study.

Permission (×4):	
[camera]	to scan (your) documents.
[location]	to search for gas stations (near you).
[microphone]	to send voice messages to (your) contacts.
[storage]	to attach photos to (your) posts.
Perspective & Politeness (×5):	
[user-demanding]	You need to allow...
[user-polite]	Please allow...
[app-demanding]	We need...
[app-polite]	We would like...
[objective]	Camera permission is needed...
Functionality (×2):	
[specific]	...to scan (your) documents.
[vague]	...(for the app) to work correctly.
Articulation (×2):	
[positive]	...to scan (your) documents.
[negative]	Without this permission, you cannot...
Additional Information (×4):	
[none] _	
[guarantee] We do not collect or transfer any personal data outside your phone.	
[control] You can change permissions from device settings anytime.	
[more] For more information, see the privacy policy on our website.	

ations, and study models upfront, minimizing biases and ensuring robust findings. For more details, please see our preregistration on the Open Science Framework [40].

5.4.2 Rationale Building Blocks for the User Study

The linguistic features from manually coded real-world rationales informed the rationale building blocks for our user study. Each rationale in the study includes a fundamental building block and may include one optional building block. This limitation restricts the number of phrases in each rationale, aiming to balance the systematic testing of various rationale combinations with participants' cognitive processing limits [32]. This design choice also mirrors the brevity observed in rationales from our exploratory analysis (see Section 5.3). Table 5.1 presents the phrasings for each rationale building block utilized in the study. When multiple options were available for a block, we prioritized the most common choice. For instance, for the [user-demanding] block, we selected *"you need"* over less common options like *"you must."* Depending on the building blocks, each rationale is tied to a permission, adopts a certain perspective, includes clear or vague functionality, is phrased either positively or negatively, and can incorporate additional information. Furthermore, the functionality can also be user-centered, such as *"to scan your documents,"* which is paired with rationales phrased from the users' perspective.

Combining the rationale building blocks resulted in a vignette experiment with five

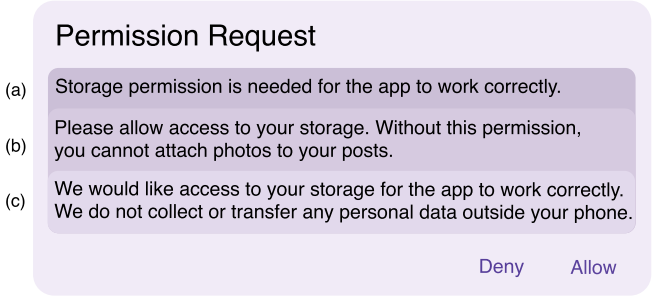


Figure 5.11: Example of three distinct rationale sentences consisting of different building block combinations. (a) objective, vague, positive, none, (b) user-polite, specific, negative, none, (c) app-polite, vague, positive, guarantee.

dimensions (see Table 5.1), with 320 unique vignettes/rationales ($4 \times 5 \times 2 \times 2 \times 4$), 80 per permission type. We presented each participant with four randomly selected rationales, each on a different permission, in random order. We also ensured an even distribution of participants among the five dimensions. Figure 5.11 shows example rationales for three different vignettes.

5.4.3 Procedure

The study was conducted as an online experiment via the survey software Qualtrics. Upon granting their consent, participants were provided with a brief introduction to the study procedure. The central part of the study was then carried out and repeated four times. In each iteration, participants encountered a randomly selected rationale from a pool of 80 rationales per permission. During this phase, participants were instructed to carefully read the presented rationale and indicate their decision to grant or deny the corresponding permission. Afterward, participants were reminded of their previous decision on a separate screen. They were asked to evaluate their decision. Lastly, participants were requested to provide demographic information. The study procedure and measurements were refined based on insights from a pilot study involving 5 participants. To learn more about the study procedure, please consult Appendix 9.2, and for further details on measurements, refer to Section 5.4.5.

5.4.4 Recruitment and Incentives

Participants were recruited using Prolific [128], ensuring a balanced sample of male and female participants. Each participant received £2 for completing the 10-minute survey (£12.00/hour). To be eligible, participants needed to be at least 18 years old, fluent in English, and regular users of mobile phones. We included participants from different mobile operating systems, like Android and iOS, as linguistic features can influence any user regardless of their OS.

To determine the optimal sample size for our study, we employed Monte Carlo simulations of the relevant multilevel models [75]. Without effect size estimates derived from prior research or meta-analyses, we assumed a standardized effect size of 0.25,

which falls within the range of a medium to large effect size [104, 51]. This choice of effect size is particularly appropriate as the study's objective is to derive practical recommendations for app developers, necessitating a difference that holds practical relevance. Additionally, our sample size's boundaries were guided by increasing the sample in multiples of full presentations of all available rationales (e.g., 320 → 640 → 960) to achieve an equal distribution between rationales. Given these boundaries, our simulations indicated that a sample size of 960 participants could detect a fixed effect of 0.25 with a power of at least 0.8 while maintaining an alpha error probability of 0.05.

In total, we gathered data from 980 participants. To maintain data quality, we followed recommendations from the literature for filtering out careless responses [164]. We excluded 18 individuals who self-reported to not use their data, one person due to missing data, and another person because they completed the task three times faster than the median speed of the sample. Our final sample consisted of 960 participants, 49.8% of whom identified as female and 48.5% as male. Additionally, 16 participants identified with other self-reported genders or chose not to disclose. Participants had a mean self-reported age of 32 ($SD = 10.8$ years). Most participants attended college, with 50.3% earning an undergraduate degree, 17.9% holding a graduate degree, and 17.1% not completing their studies. Regarding smartphones, 60.3% used Android, 38.1% used an iPhone with iOS, and 1.6% used a Windows phone, which is comparable to the worldwide mobile OS marketshare [57].

To ensure our study represents a worldwide viewpoint and to increase the relevance of our findings across cultural boundaries, we recruited participants from different countries as long as they spoke fluent English. Consequently, our final international sample included participants from a wide range of geographic regions spanning multiple continents, as facilitated by Prolific (45% from Europe, 37% from the Americas, 13% from Africa, 4% from Oceania, and 1% from Asia), representing a total of 31 different countries. Refer to Appendix 9.1 for a detailed breakdown of participant countries.

5.4.5 Measurements

In our study, we utilized a range of measurements, which are explained in this section and are also available in the questionnaire provided in Appendix 9.2.

Decision & Decision Evaluation. Participant's decision was assessed with one item asking them to choose between *"allow"* or *"deny"*, which resembled the choice in the presented app screenshot of the rationales.

To assess users' evaluation of their decisions, we used an adapted version of the Decision Evaluation Scales (DES), which was successfully used in this context before [P1]. The scale was initially adopted from the field of health psychology [148], where it was designed to evaluate patients' decision to uptake or refuse a treatment choice, which is analogous to users' decision to grant or deny a permission request. Additionally, the DES allows the investigation of multiple essential dimensions of the decision evaluation, such as (1) whether users received sufficient information to make an informed decision, (2) their satisfaction with the decision, and (3) their perceived control over the decision.

For the user study, we extended the scale used in the related work by adding one

additional item to each subscale. In this way, we aimed to increase the reliability of the assessment [7], capture a broader perspective on the underlying construct, and prevent censored scale averages due to low item difficulty. To create additional items for the scale, we constructed a set of five new items for each subscale, which were then subsequently rated by a sample of nine domain experts (four behavioral scientists and five information security experts). The three items with the highest agreement were then added to the user study. The scale items, the results for multilevel internal consistency, and factor loadings for all items from confirmatory factor analysis are given in Appendix 9.3. All the subscales of the DES were measured on a seven-point rating scale, with scores closer to 7 indicating greater agreement with the item and scores closer to 1 indicating greater disagreement.

Person Level Measurements. To account for individual variations, besides gathering demographic data (like gender, age, educational level, and users' mobile OS), we also assessed participants' privacy concerns and their past privacy experiences. Previous research has indicated that these factors can influence users' decisions regarding runtime permission requests [P1, 24, 174].

5.4.6 Ethical Considerations

The study was approved by our institution's Ethics Review Board. Data collected via Prolific and Qualtrics were treated sensitively, with personal identifiers separated. At the start of the study, participants received precise details about the purpose of the study and the data being collected. We ensured participants understood how their data would be used while allowing them to withdraw their participation at any time.

5.4.7 Model Construction

We used linear multilevel models to test whether rationale phrasings impacted users' decisions and conducted all analyses in R version 4.2.2 (R Core Team, 2023). As part of our data preparation, we computed scale means for measurements with multiple items, including informed decision, decision satisfaction, decision control, prior privacy experience, and privacy concerns. Additionally, we standardized all user-level predictors (age, prior privacy experience, and privacy concerns) by using grand mean centering. Categorical predictors were coded using treatment coding, with the reference groups as follows: perspective & politeness (objective), articulation (negative), functionality (vague), and additional information (none). We deviated from our preregistration for permissions and chose to treat them as random effects rather than fixed effects with difference coding. This change did not substantially affect the impact of other variables but allowed us to investigate the variation between permissions more closely.

We took a step-by-step approach to simplify our modeling process and ensure consistency with recommendations from prior research [79]. We used maximum likelihood estimation for all models to make them comparable. We built and tested the models as follows: (1) In the first step, we started with a simple regression model. Next, we expanded it to a random intercept model, considering permission and user as random effects. (2) For the second step, we introduced control variables, specifically prior privacy

Table 5.2: The final multilevel models.

	Decision *Odds Ratio* (*std.* β)	DES Inform β (*std.* β)	DES Satis β (*std.* β)	DES Control β (*std.* β)
(Intercept)	1.76 (1.76)***	-1.01 (-0.61)***	1.68 (0.19)***	0.27 (-0.01)***
Privacy Concerns	0.71 (0.71)***	0.12 (0.11)***	0.09 (0.11)***	0.03 (0.03)
Prior Experience	0.89 (0.88)**	-0.02 (-0.02)	-0.08 (-0.08)***	-0.17 (-0.16)***
Decision Grant	–	1.08 (0.75)***	-0.30 (-0.28)***	-0.15 (-0.12)***
Perspective & Politeness (ref: objective)				
user-demanding	0.87 (0.87)	0.05 (0.03)	0.00 (0.00)	-0.03 (-0.02)
user-polite	0.86 (0.86)	0.01 (0.01)	0.04 (0.04)	0.00 (0.00)
app-demanding	0.80 (0.80)	-0.00 (-0.00)	-0.04 (-0.04)	0.02 (0.02)
app-polite	0.71 (0.71)**	0.06 (0.04)	0.00 (0.00)	0.07 (0.06)
Functionality (ref: vague)				
specific	1.12 (1.12)	0.16 (0.11)***	0.07 (0.07)*	0.03 (0.03)
Articulation (ref: negative)				
positive	1.03 (1.03)	-0.02 (-0.01)	-0.09 (-0.08)**	-0.02 (-0.02)
Additional Information (ref: none)				
guarantee	1.43 (1.43)***	0.24 (0.16)***	0.06 (0.05)	0.17 (0.15)***
control	1.28 (1.28)*	0.06 (0.04)	-0.01 (-0.01)	0.13 (0.11)**
more	1.07 (1.07)	0.11 (0.08)*	-0.07 (-0.07)	0.05 (0.05)
Marginal R^2	0.050	0.141	0.037	0.029
Conditional R^2	0.177	0.414	0.381	0.499

Models were fitted with the Restricted Maximum Likelihood estimation. The coefficients for Decision are shown as odds ratios, where values <1 indicate that the likelihood of granting permissions is lower than the likelihood of denying the permission and values >1 indicate that the likelihood of granting the permission is higher. *std.* β = *standardized* β. $^{*}p < .05$, $^{**}p < .001$, $^{***}p < .0001$. Decision coding: 0 = *deny*, 1 = *allow*. N: 960_{User}, $3840_{\text{Rationale}}$.

experience and privacy concerns. For the DES models, we also incorporated participants' decisions as a control variable because the outcome (i.e., granting or denying permission) could affect users' comfort level with their choices. (3) Progressing to the third step, we added the variables of interest. These included perspective & politeness, articulation, functionality, and additional information. (4) In the fourth step, we tested adding interactions between the variables of interest. However, including these did not enhance the model fit. For more details about the model-building process and fit criteria, see Appendix 9.4. The final models were recalculated using Restricted Maximum Likelihood Estimation, which leads to a more conservative and less error-prone estimation of the parameters [79]. Table 5.2 shows the final model for each outcome variable.

5.4.8 Results

Next, we present the results of our user study and discuss their implications in Section 5.5.

Effect of rationale building blocks. Below are the outcomes of the effects of the five rationale building blocks.

Perspective & Politeness: When rationales were phrased politely from the app's perspective (e.g., *"we would like"*), it had an interesting effect. The likelihood of granting permissions decreased ($odds\ ratio = 0.71$, $std.\ \beta = 0.71$, $p = 0.003$). However, whether the rationale was phrased from different perspectives and had a polite or demanding tone did not influence how participants perceived their decision.

Functionality: Rationales that explained why permission is needed had a positive impact on participants, making them feel more informed about their decision ($\beta = 0.16$, $std.\,\beta = 0.11$, $p < 0.001$) and more satisfied with their choices ($\beta = 0.07$, $std.\,\beta = 0.07$, $p = 0.018$). However, this did not significantly increase the likelihood of granting permissions or make participants feel more in control of their decisions.

Articulation: The tone used in the rationale also played a role. Positively phrasing the rationale, highlighting the benefits of granting permission, decreased decision satisfaction ($\beta = -0.09$, $std.\,\beta = -0.08$, $p = 0.005$). However, this positive phrasing did not affect the decision itself, the perception of being informed, or the sense of control over the decision.

Additional Information (guarantee): Guaranteeing that the permission will not be misused had a notable impact. It significantly increased the likelihood of participants granting permission ($odds\ ratio = 1.43$, $std.\,\beta = 1.43$, $p < 0.001$). Furthermore, participants felt more informed about their decision ($\beta = 0.24$, $std.\,\beta = 0.16$, $p < 0.001$) and more in control ($\beta = 0.17$, $std.\,\beta = 0.15$, $p < 0.001$). Still, it did not have a direct effect on decision satisfaction.

Additional Information (control): Informing participants that they can change their decision at any time increased the likelihood of granting permission ($odds\,ratio = 1.28$, $std.\,\beta = 1.28$, $p = 0.018$). It also made participants feel more in control of their decisions ($\beta = 0.13$, $std.\,\beta = 0.11$, $p = 0.002$). However, it did not impact participants' perception of their decision being informed or their satisfaction with the decision.

Additional Information (more): Interestingly, adding the phrase *"For more information, see the privacy policy on our website"* did not provide additional information but increased participants' perception of being informed ($\beta = 0.11$, $std.\,\beta = 0.08$, $p = 0.047$). Other variables, such as the decision itself, satisfaction, and control, remained unaffected.

Effect of the generic rationale. Our design included a generic base rationale, which consisted of a single negatively articulated phrase from the objective point of view. This phrase described a vague functionality, was neither polite nor included any additional building blocks (e.g., *"Without storage permission, the app cannot work correctly."*). The effect of this rationale can be investigated by interpreting the intercepts of our dependent variables. In this context, the baseline zero means either denying the permission (Decision) or represents the average score of 4 across the three decision evaluations (DES Inform, DES Satis, DES Control). On all four variables, our results showed that participants deviated significantly from this baseline in terms of declining the permission request or feeling indifferent when presented with the generic rationale. Even with the generic phrasing, participants were more likely to grant the permission ($oddsratio = 1.76$, $std.\beta = 1.76$, $p < 0.001$). They felt less informed ($\beta = -1.01$, $std.\beta = -0.61$, $p < 0.001$), but still in control ($\beta = 0.27$, $std.\,\beta = -0.01$, $p < 0.001$) and satisfied ($\beta = 1.68$, $std.\,\beta = 0.19$, $p < 0.001$) with their decision.

Effect of Other Variables. We also examined the effects of other variables related to individual differences or known to influence permission decisions from prior research.

Privacy Concerns: Participants' privacy concerns had a multifaceted impact on users' decisions and perceptions. Firstly, higher privacy concerns were associated with a decreased likelihood of granting permissions, indicating that individuals with privacy concerns were less inclined to provide access ($odds\,ratio = 0.77$, $std.\,\beta = 0.71$, $p < 0.001$). Conversely, higher privacy concerns had a positive influence on participants' perception of making an informed decision ($\beta = 0.12$, $std.\,\beta = 0.11$, $p < 0.001$) and their satisfaction with the decision ($\beta = 0.09$, $std.\,\beta = 0.11$, $p < 0.001$). However, privacy concerns did not significantly impact participants' perception of control over their decisions.

Prior Privacy Experience: Examining participants' previous encounters with privacy-related experiences provided the following insights. Participants with more prior interactions with privacy issues were less likely to grant permissions ($odds\,ratio = 0.89$, $std.\,\beta = 0.88$, $p = 0.006$). This trend also extended to participants reporting lower overall satisfaction with their decisions ($\beta = -0.08$, $std.\,\beta = -0.08$, $p = 0.001$) and a diminished sense of control over their choices ($\beta = -0.17$, $std.\,\beta = -0.16$, $p < 0.001$). Prior privacy experience did not affect the perception of making an informed decision.

Permission Decision: Analyzing participants' decisions revealed that granting permission increased their sense of being informed ($\beta = 1.08$, $std.\,\beta = 0.75$, $p < 0.001$) but came with trade-offs. It lowered both decision satisfaction ($\beta = -0.30$, $std.\,\beta = -0.28$, $p < 0.001$) and the sense of control ($\beta = -0.15$, $std.\,\beta = -0.12$, $p < 0.001$). In essence, while granting permission made participants feel informed, it reduced satisfaction and control.

Age, Gender, Educational Background & Mobile OS: Previous studies in the field have investigated age, gender, educational background, and mobile OS as possible confounding variables [24, 28]. To investigate the effects of these variables on our main outcomes, we compared the models with these added control variables to the models reported above. For the control variables, we excluded answers with only a few observations (e.g., the "other" category in educational background). We refitted the models reported above on the reduced dataset ($N_{reduced} = 929$) to allow a numeric comparison depending on the common fit measures. Inspecting these comparisons, only the models for decision ($\chi^2(5,\ N = 929) = 12.72$, $p = 0.029$) and informed decision ($\chi^2(5,\ N = 929) = 12.524$, $p = 0.028$) improved significantly over the study models.

For these two models, none of the independent variables' effects were decisively affected by adding the additional control variables to the model. For decision, we found that participant's odds of granting permission were significantly increased if they held an undergraduate degree compared to a graduate degree ($odds\,ratio = 1.32$, $std.\,\beta = 1.32$, $p = 0.019$) and decreased if they were using iOS compared to Android ($odds\,ratio = 0.79$, $std.\,\beta = 0.79$, $p = 0.010$). Additionally, participants who identified as female felt they had made a slightly less informed decision ($\beta = -0.15$, $std.\,\beta = -0.11$, $p = 0.008$) compared to those who identified as male. We did not find a significant effect of age.

Variation between clusters. A considerable proportion of the variance in the models was explained by differences within participants and permissions instead of only the fixed factors. This is expected as evaluating the decision to grant permission is a complex cognitive process most likely influenced by many different characteristics of the participants (e.g., the propensity to make a decision) and the provider of the rationale (e.g., the app's trustworthiness). Although we did not measure specific characteristics of the participants in this regard, our hierarchical analysis accounted for these differences in clusters of participants and permissions and uncovered that the base effect of the rationales mainly varies within participants, ranging from an intercept variance of $\tau_{00\_\text{Satisfaction}} = 0.40$ to $\tau_{00\_\text{Control}} = 0.64$, while there was only a small variance based on the permissions ($\tau_{00} = [0.00; 0.04]$).

5.5 Discussion

App developers have many choices when crafting rationales, including prompt styles and different phrasings. Despite the availability of various guidelines on creating rationales [117, 11, 66], our empirical analysis revealed significant diversity in developers' approaches. Additionally, studies in other domains have shown how linguistic variations in phrasing can affect various user decisions [58, 5, 37, 159, 90, 141]. Given these factors, it is essential to understand how the diverse phrasing of rationales can influence users' permission decisions and, consequently, their privacy. Focusing on existing guidelines, this discussion will show their adoption in our dataset, compare available recommendations with our findings, and distinguish our research from other studies on the trustworthiness of rationales. We will also provide directions for implementing our recommendations in future research.

5.5.1 Do Real-World Rationales Follow Guidelines?

Guidelines from Android [66], iOS [11], and NN/g [117] recommend the following for phrasing rationales: provide specific functionality, avoid passive voice, and include user-related features. In our dataset, we found that the majority of rationales adhere to these recommendations. For instance, more than half of the rationales we inspected explain why permission is necessary, specifying a particular functionality ($58\%^{\text{m}}$ $69\%^{\text{s}}$). Furthermore, many app developers avoid using passive voice (objective perspective in our analysis) in their rationales ($73\%^{\text{m}}$ $70\%^{\text{s}}$). However, only a small fraction of rationales emphasize the benefits to the user when granting permission ($18\%^{\text{m}}$ $17\%^{\text{s}}$), such as *"so you can make video calls, "* or *"to share pictures with your friends."* Even though there is relatively high compliance on the side of developers, many real-world examples phrase their rationales differently or opt to include additional details.

5.5.2 Influential Building Blocks Within and Beyond Guidelines

Every user possesses distinct privacy concerns and preferences. We tried to apprehend these nuances by capturing how users perceive and assess their permission choices, regardless of whether they grant or deny permissions. Our assertion is that users should feel well-informed, satisfied, and in control of their decisions, aligning with

their individual privacy preferences within a given context. In summary, we provide actionable recommendations for developers based on the insights from the user study. These recommendations can serve as a more granular extension of the existing guidelines.

Provide specific functionality and phrase it negatively. In line with available guidelines, we found that clearly explaining why permission is required by specifying a functionality improves users' perception of having made an informed decision and increases their satisfaction with that decision. Notably, users tend to be more satisfied with their decision when permission requirements are presented in a negative context, such as "Without this permission, {certain app functionality} cannot be used." This negative phrasing appears to be more straightforward for users, enhancing their satisfaction with the decision-making process compared to a positive phrasing, such as "This permission is used for {certain app functionality}."

We also found that supplementary information blocks had a positive impact on users. However, because space for rationales is limited, we recommend that app developers prioritize these blocks based on their effectiveness, using space as available.

Assure users what the permission will not be used for. The first and most influential addition is including a guarantee that permission will not be misused. Our study found that this addition significantly enhances users' perception of making an informed decision, empowers them to feel more in control, and increases grant rates. Users were influenced by such assurances even without concrete proof, possibly due to perceived transparency from the app developers or a misbelief that app distribution platforms prevent fraudulent permission requests. As such misconceptions pose a risk to users, it is essential that developers include this building block only if the app genuinely upholds these promises and provides a legally binding privacy policy. Furthermore, we recommend that the truthfulness of this statement should be verified using information from the rationale during app vetting. In fact, the mandated presence of such useful information supports the vetting process, as detailed in Section 5.5.4.

Highlight the reversibility of permission decisions. The second addition we examined is reminding users that they can modify their permission choices, articulated as "You can change permissions from device settings anytime." This phrase enhanced users' perception of control and increased their likelihood of granting permission. This effect aligns with the control paradox, which posits that the perception of control increases the likelihood of disclosing sensitive information [25]. It may also stem from the notion that users feel less apprehensive when they have a sense of control over their decisions, similar to being the driver of a car rather than a passenger. This finding is consistent with prior research, which shows that users are more likely to grant permission when aware of the option to change their decision later [24]. However, other studies indicate that users rarely exercise this option [142, 101]. Therefore, highlighting the reversibility of permission decisions could increase user awareness and positively influence their perception of the decision. However, adding this information to a rationale is not enough; we must also translate this sense of control into actionable steps. Given the benefits of knowing that decisions can be revoked, future research could focus

on effectuating this sense of control, such as nudging users to review their previous permission choices [174, 8] or providing more fine-grained permission controls, akin to one-time granting of some permissions on Android [64] and iOS [9].

Provide supplemental information. The third addition is the phrase "For more information, see the privacy policy on our website." Although this phrase does not provide specific information about the purpose of the permission, it enhances users' perception of being informed. This outcome may result from the impression of users that the app developer values transparency, offers supplementary information, and complies with legal requirements. These factors collectively foster a sense of being well-informed among users, despite the absence of explicit details or direct links to additional material in the rationale. However, it is possible that the users in our study, who were prompted to imagine a real situation, just acted on the assumption that in a real-life situation, they would have been able to obtain more information about the app and its features. If not, this finding is somewhat worrisome because no further informational value was added to the rationale, but still, users felt better informed. Nevertheless, in practice, we recommend adding a link to supplemental information for users who need more information to decide. Going a step further, our results indicate that users may assume any additional information, even just a hint to a privacy policy, is verified and trustworthy. It appears that users delegate the vetting process, to other users or Google. Therefore, we see an opportunity for future research to shift this implicit user trust to explicit verification. Ideally, rationales should distinguish between provided and verified privacy policies, highlighting and differentiating them.

5.5.3 Revisiting Rationale Guidelines

We found that not all available guidelines influenced our outcome variables as expected. For example, the specificity of the provided functionality had no significant effect on users' permission decisions. Whether specifying the use of permissions has no impact or only a minor effect on grant rates remains undetermined, as our study design may not have detected these nuances. However, our observation aligns with previous research, which has shown that any explanation provided by developers tends to increase grant rates, regardless of the explanation's meaningfulness [156].

Additionally, contrary to guidelines advising against passive voice, we found that using the passive voice (the objective perspective) in rationales did not have a detrimental impact. In fact, in some cases, rationales written in the passive voice had a higher likelihood of being granted compared to those phrased politely from the app's perspective, such as "we would like your permission." Although not directly examined in our study, we suspect that polite language might make users perceive the permission as optional, resulting in lower grant rates. Moreover, we did not find a significant effect of adding user-related features to the rationale.

In conclusion, this does not mean that these guidelines do not contribute to the overall readability or user comprehension; it simply means that we did not find a significant impact on our main outcome variables, which we consider relevant indicators of improving the overall user experience. Thus, our results suggest that existing

guidelines can still be enhanced, for example, by adding supplementary information. Additionally, we found that the perspective of a rationale is less critical than previously thought. Finding further gaps might also be an interesting question for future research.

5.5.4 Differentiating Between Usability and Trustworthiness

In this work, our efforts prioritize enhancing the usability of rationales, which is distinct from addressing their trustworthiness and verification. This differentiation mirrors similar distinctions in research concerning other self-reported privacy instruments such as privacy policies, Privacy Nutrition Labels, and Google's Data Safety Section. While some studies concentrate on enhancing the usable security aspects [98, 176, 177, 33], others focus on improving and assessing the trustworthiness of these instruments [89, 88, 91, 53, 173, 83, 151]. Due to this distinction, it is important to note that malicious app developers could potentially exploit the above phrasing strategies, which enhance the usability of app rationales, to the user's disadvantage. Consequently, ensuring the trustworthiness of rationales remains an important question that needs to be addressed by parallel works. For example, app vetting can use rationales as reference points in analyzing and classifying app behavior. Ultimately, both aspects are indispensable: a usable yet unverified rationale holds no more value than an unusable verified rationale. If rationales are acknowledged as effective privacy declarations by end-users, it becomes imperative to mandate app developers to provide them in a structured format. This enhances user comprehension and allows for the development of solutions that can analyze privacy compliance on a large scale, much like privacy nutrition labels [177].

5.5.5 Future Directions for Guideline Adoption

Having discussed the available guidelines and provided more nuanced recommendations, the question of how to best assist developers in implementing these guidelines remains open. On one hand, we believe that rationale guidelines should remain recommendations to guide app developers toward best practices without being enforced, allowing developers the freedom to create rationales aligned with their corporate identity. On the other hand, given that the guidelines proposed in this work emphasize the phrasing of rationales, there is significant potential for developing a tool for modular rationales. Such a tool would allow developers to utilize rationale building blocks, enabling them to assemble customized rationales that suit their specific needs. This approach could reduce the burden on developers in crafting rationales while still giving them the flexibility to design the UI of rationale messages that reflect their corporate identity. This tool could also be used for iOS rationales which, in specific cases, can be provided in custom pre-alert screens [11] (as in Android) in addition to a purpose string that is integrated into the permission request dialog.

5.6 Threats to Validity

Our user study is subject to methodical deliberations, which constrain its scope in terms of certain forms of validity. In pursuit of high internal validity, we implemented a rigorous experimental design that held all influencing factors constant, thus increasing

the likelihood that the results reflect only the actual effects of rationale building blocks. However, this approach required presenting permission requests through vignettes rather than asking users to install an app on their device that monitors the handling of permission requests and subsequently poses questions. Limited external validity is inherent in every vignette study, as it prompts users to imagine the situation rather than being in it, thus not fully reflecting real-world permission request handling. However, continuously monitoring user's device usage also raises significant ethical concerns. In our case, no prior studies existed that would justify infringing on a user's privacy without knowing if rationale phrasing would have a measurable effect on their decisions. Instead, our approach allowed us to test numerous possible variations under controlled conditions, whereas a field study would rely on a sample of rationales from incidental app installations, which also limits its generalizability. Nevertheless, we still displayed the rationale in the design of a genuine app prompt on a smartphone to enhance immersion and realism. Consequently, we contend that our chosen study design was the most appropriate under the given circumstances.

Additionally, our use of a repeated measurement design may have caused participants to suspect the research purpose to some extent when encountering the second rationale. To mitigate potential sequence effects, we employed randomization by assigning random permission orders for each participant and presenting them with a randomly selected set of rationales. Additionally, we explicitly instructed users to assume that every prompt represented a new app installation. While these measures counteract common sequence effects, they cannot completely eliminate the possibility of user boredom or careless responses due to repetition. Therefore, we conducted rigorous data analysis to identify indications of such issues and handled them appropriately.

Furthermore, it is important to note that we did not explore the impact of different rationale designs, which could potentially have a significant effect on user perception. Our choice to focus on a single design was primarily driven by feasibility considerations, as discussed in Section 5.4.1. Investigating multiple design parameters alongside the rationale variations would also have complicated our statistical models and might have led to issues with multicollinearity. Therefore, we opted for a consistent rationale design that was commonly observed in the rationales we examined. Nonetheless, we encourage further research on this topic, building upon the comprehensive investigation of design variations reported in this study.

On a final note, while the results of our study demonstrate the influence of rationale variations on users, it is important to recognize that they have not been field-tested, where other contextual factors may play a significant role. To accurately anticipate the real-world impact of our recommendations, it is essential to remember that human behavior is rarely influenced by a single, straightforward cause, and rationales are just one factor among many. Thus, when applying and interpreting these results, it is crucial to keep in mind that we are dealing with subtle nuances in language and content rather than dramatic changes in design.

5.7 Conclusion

In this work, we extensively examined real-world rationales in the context of mobile app permissions. Through this investigation, we uncovered diverse building blocks and design elements of rationales. Our user study, involving 960 participants, unveiled the impact of phrasing on users' permission decisions and their assessment of those decisions. By aligning our findings with established recommendations and guidelines, we extracted valuable insights, offering actionable recommendations for app developers to enhance user experience through more effective rationale crafting. Our work underscores the importance of well-considered phrasing of rationales and extends an invitation for future research. Subsequent investigations could explore additional dimensions, emphasizing refining the design and overall user experience of rationales. Furthermore, establishing rationales as a usable and standardized instrument can yield enhancements in other areas, such as app vetting, including the validation of rationale messages and Google's Data Safety Section.

6
Web Rationales

Permission Rationales in the Web Ecosystem: An Exploration of Rationale Text and Design Patterns

6.1 Motivation

Modern web applications are becoming increasingly feature-rich and interactive, offering users a more dynamic and personalized online experience by utilizing device resources like cameras, microphones, push notifications, and geolocation data. However, to harness these capabilities, websites must often first ask users for permission through browser permissions prompts. While these permissions are critical to enable key features safely, they also introduce a significant burden for users. When confronted with permission prompts, users must make informed decisions about which capability accesses to allow and which to deny. Deciding wrongly can have negative security and privacy consequences, depending on the capability in question. In addition, prior work has shown that websites often ask for permissions in inopportune moments, making these requests annoying and lacking context [71].

Permission prompts on mobile platforms, particularly on Android, have been a frequent focus of security and privacy research studies over the years [P1, 28, 100, 170, 126, 24, 127, 48, 86, 167, 73, 109, 155, 157, 160, 97, 23, P2, 96, 101, 171, 119, 8, 158, 156, 142, 165, 174, 102, 103, 115, 116]. Unlike mobile apps, which are distributed through app stores with strict guidelines, websites are delivered dynamically via web browsers. Mobile app stores also impose specific user experience (UX) requirements and guidelines for permission requests [10, 62], the web operates with significantly less centralized oversight. Web developers can trigger permission prompts at any point, even right after the page finished loading, and without following best practices [74].

The research community has only recently begun exploring *web* permissions, primarily focusing on user experiences with permission prompts [71], in particular for push notifications [20, 72]. Previous studies have examined user perceptions of annoyance or interruption, the ease or difficulty of decision-making, and the presence of contextual information on desktop platforms. However, the way browsers display these prompts is just the tip of the iceberg. The broader context, including how websites present permission requests and the rationales provided to users before and after prompts, remains largely unexplored in terms of both scope and impact.

Permission rationales on the web are explanations added to webpages to clarify why certain capabilities are required, providing essential context for permission requests. Research has consistently shown that offering contextual information significantly impacts user interactions with permission prompts [71, P1, P2, 157, 160]. In the Android ecosystem, studies have highlighted not only the importance of rationales but also the diverse ways contextual information is presented, demonstrating their critical role in shaping user decisions [P1, P2, 157, 160]. However, despite the acknowledged importance of rationales in shaping user responses to web permission prompts [71], we still lack detailed information on the variety of web rationale texts and designs, methods to automatically trigger and detect them, the prevalence of websites using rationales, and the effects of different rationale patterns and design choices on user decision-making regarding permission prompts.

6.2 Contribution

In this work, we conduct the first systematic and comprehensive study of web permission rationales on desktop platforms, a largely overlooked aspect of the web permission ecosystem. Our research focuses on (i) systematically exploring and collecting webpages that feature permission prompts, (ii) automatically detecting and classifying rationales, and (iii) thoroughly analyzing various text and UI attributes of these rationales to begin understanding their impact on user actions and sentiment toward permission prompts.

Starting with 770K URLs from Chrome telemetry, we performed automated, interactive web crawling. We collected snapshots of webpages that request the most common permissions-gated web APIs, i.e., notifications, geolocation, camera, and microphone. We considered both screenshots for rationale UIs and the DOM of the page [168] for rationale text. As a result, our crawler successfully captured snapshots for 739K reachable URLs and triggered permission prompts on over 20% of the visited webpages.

We detected and manually confirmed 3.6K unique text rationales using a robust machine-learning pipeline. In addition, we semi-automatically compiled a dataset of 749 distinct rationale UIs. We observed that 85K webpages in the wild use one of the 3.6K unique rationale instances. We found that the most prevalent rationales belong to 10 libraries. Then, we undertook a qualitative analysis to characterize rationales, considering various aspects, including message tone, encouragement, message content, functionality necessity for text and layout, position, elements, and timing for UI.

After analyzing rationales to extract their attributes, we conducted an exploratory analysis to study how these elements impact users' decisions to grant, deny, dismiss, or ignore permission prompts, again using Chrome telemetry and user sentiment data. We applied regression models to extract 10 key effects. Among others, we find that any rationale message, regardless of tone, significantly boosts grant rates and reduces dismiss and deny rates, with positive tones increasing grant rates by 18%. Additionally, we find that UI design elements can have an even higher impact. For example, overlays before or alongside a prompt had the most substantial impact on grant rates (+41%), followed by fullscreen rationales (+33%). When it comes to user sentiment, dialogs and text rationales were associated with increased user annoyance, particularly when shown before and after browser prompts.

In summary, we make the following main contributions:

- We create the first (semi-)automated approach to detect and study web permission rationales at scale. We instantiate our approach on a set of 770K webpages, processing more than than 6M unique text snippets. As a result, we create a comprehensive dataset of 3.6K manually-vetted and unique rationale text and 749 rationale UIs on the desktop web.
- We estimate the prevalence of web permission rationales, focusing on push notifications, geolocation, camera and microphone permissions, identifying ∼85K webpages that use a custom or library-provided rationale. We find 10 libraries that have the most prevalent rationales–with the top three being OneSignal [120], iZooto [82] and Smart Push [78]– and create 32 code signatures to detect their use on webpages.

– We conduct a qualitative analysis of permission rationales, examining both text and UI. For the rationale text, we extracted attributes across four dimensions: message tone, encouragement, content, and functional necessity. For the rationale UIs, we identified attributes spanning three dimensions including layout elements, position and timing. We used these attributes to characterize web rationales, identifying 18 common rationale text patterns and 8 common UI patterns.
– We study the impact of rationale attributes on user behavior and sentiment towards permission prompts, cross-referencing webpages with coded rationales against Chrome telemetry and user sentiment data, and extract nine key insights.

6.3 Methodology

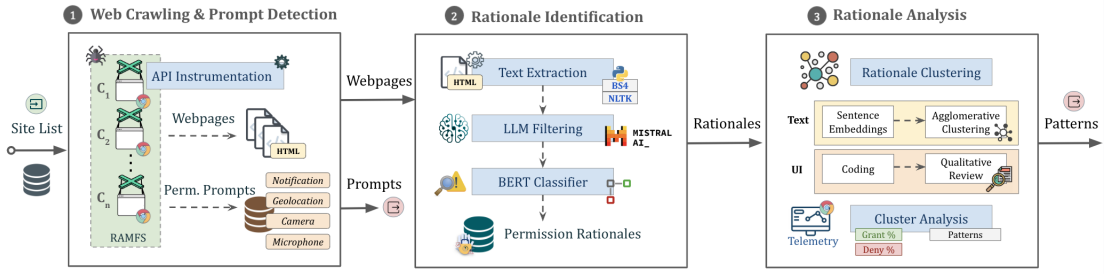


Figure 6.1: Overview of our methodology.

To address our research questions, we follow the methodology outlined in Figure 6.1. We begin with *"Web Crawling & Prompt Detection"*, where we use a JavaScript-enabled web crawler to navigate a list of seed URLs. During each visit, the crawler captures webpage snapshots and screenshots, while also monitoring the page to collect any permission prompts that appear. Next, in *"Rationale Identification,"* we extract distinct text snippets from webpages. These snippets are processed through Large Language Models (LLMs) to identify those that pertain to permission-protected concepts, such as access to a camera or microphone. Using this filtered data, we manually curate a ground-truth dataset for permission rationales. This dataset is then used to train a BERT classifier to identify rationales. As a result, we build a comprehensive catalog of rationales. Finally, in *"Analysis of Rationale Text & UI patterns,"* we apply both automated clustering and manual coding to identify rationale patterns. This step incorporates both quantitative analysis of textual content and qualitative examination of UI elements of rationales. Finally, in *"Exploring the Effect of Rationales on User Decision-Making,"* we evaluate the impact of these rationale patterns by comparing permission grant and deny rates based on user activity data from Chrome telemetry.

6.3.1 Web Crawling & Prompt Detection

To answer RQ1, we developed a Chromium-based web crawler using Puppeteer and the DevTools Protocol (CDP) to capture webpage snapshots, simulating a desktop browser. The crawler loads an initialization script that modifies JavaScript permission

Table 6.1: Permission prompts seen by unqiue desktop/mobile clients.

Platform	🖵		📱		**Total**	
Permission	**Pages**	**Sites**	**Pages**	**Sites**	**Pages**	**Sites**
Geolocation	192,728	46,450	272,007	47,210	464,735	93,660
Notification	263,835	30,523	424,264	35,054	688,099	65,577
Microphone	11,046	4,054	11,217	3,898	22,263	7,952
Camera	9,336	3,863	18,407	6,835	27,743	10,698
Total	476,945	77,086	725,895	86,572	**770,349**	118,371

APIs, enabling it to monitor permission prompts in real-time. For each webpage, the crawler waits up to 30 seconds for the page to fully load, then collects the client-side code, a Document Object Model (DOM) snapshot, and a screenshot to capture the main rationales presented in the webpage's user interface (UI).

For our analysis, we focus on telemetry from Chrome, which is known to be representative of popular websites, as demonstrated in recent research [137]. Specifically, we acquired a Chrome telemetry dataset from December 2022, comprising 770K publicly accessible URLs. Each URL corresponds to a specific webpage where at least 50 users across all platforms encountered a permission prompt within the last 28 days, as shown in Table 6.1. The URLs were sanitized to avoid exposing sensitive information.

During each webpage visit, the crawler tracks calls to permission-restricted APIs and records the permissions requested. If a permission request appears, the crawler rejects it using the CDP and captures a second snapshot to identify any secondary rationales that might appear when permission is not granted (whether ignored, dismissed, or denied). To maximize the detection of prompts and capture associated rationales, the crawler also interacts with the webpage by clicking on elements likely to trigger permission-related actions. The full list of interaction heuristics is provided in Table 10.2 of Appendix 10.1. With these heuristics, our crawler detects nearly twice as many permission prompts compared to a non-interactive approach (see Appendix 10.1.1).

6.3.2 Rationale Identification

To identify and extract rationale sentences from the crawled webpages, we followed a multi-step process involving text extraction, dataset construction, and model training.

We used BeautifulSoup [134] to parse each webpage's HTML and extract raw text, focusing on rationale sentences while excluding headers, footers, and stylistic elements. The text was deduplicated to retain unique samples for further processing.

Ground-truth dataset creation with LLM filtering. Using the unique extracted texts, we then constructed a ground-truth dataset to train a rationale classifier. Because permission rationales represent a small fraction of the vast text data, random sampling was impractical due to a low probability of identifying rationale sentences at scale. Previous research [102, 103, 116, 115] addressed a similar challenge using keyword matching, but this approach restricts collected samples to those containing predefined keywords (e.g., "camera" or "webcam") and often results in models overfitting to these

terms. To avoid this, we adopted a keyword-agnostic method, employing few-shot prompting with the Mistral-7B language model [4] to identify relevant text snippets around *permission-protected concepts* such as camera, microphone, notifications, and geolocation. This filtering substantially reduced irrelevant samples.

After filtering, we applied random sampling to the remaining data, manually labeling each sample to identify rationale sentences. This process was repeated until we had a sufficient number of positive examples. We then balanced the dataset by under-sampling non-rationale examples, creating an equal mix of positive and negative samples for training. Details of the prompt used for the model are provided in Appendix 10.4.

Large-scale rationale classification. Using the labeled dataset, we trained a BERT classifier [45], as it outperformed alternative models such as T5 in our context. The classifier was trained over 10 epochs with a learning rate of 0.0001, ensuring gradual and stable model updates. A batch size of 16 was chosen to balance computational efficiency with the need for sample diversity, and gradient accumulation steps were set to one, allowing parameter updates after each batch. A weight decay of 0.005 was also applied to improve generalization by discouraging large parameter values. These parameters were chosen based on best practices in BERT fine-tuning [1, 54, 131] and our initial experiments. Finally, we used the trained BERT classifier to categorize the unique text extracted from the webpages.

Manual review and false positive analysis. To eliminate false positives, two researchers independently reviewed the samples classified as rationales, verifying accuracy by examining the associated webpages or UIs. For cases of disagreement, a third researcher conducted an additional review, followed by a discussion among the three researchers to resolve conflicts. All reviewers were experts in web and usable security.

6.3.3 Analysis of Rationale Text & UIs Patterns

To answer RQ2, we analyzed the classification results and grouped the rationales into clusters to identify common patterns, considering both rationale text and UI components.

Rationale text patterns. For text analysis, we generated embeddings of rationale text snippets using the all-MiniLM-L6-v2 sentence transformer [46] and applied agglomerative clustering [139] to capture syntactic and semantic relationships. Through random sampling, we examined each cluster, compiling a comprehensive list of codes and attributes related to message sentiment and content until no new information emerged (saturation). We labeled the rationale samples based on these attributes, initially using one-shot prompting with GPT-4 to assign attributes, followed by a manual review for accuracy. This allowed us to group rationales with similar attributes. The results of this stage are presented in Section 6.6.1.1.

Rationale UI patterns. In the UI analysis, we performed a qualitative review of different UI designs. This included both pages where a text rationale was detected by our ML pipeline and random samples from pages with observed prompts but no

detected rationales. For each case, we analyzed screenshots captured before and after the denial of a permission prompt. If screenshots lacked the rationale (e.g., due to complex interactions missed by the crawler), we used a semi-automated approach to capture them by manually interacting with the page. After obtaining two screenshots for each rationale, two independent reviewers analyzed and coded the UI components, continuing this process until saturation. In total, we analyzed 7,413 webpages, resulting in 749 distinct UI rationales, which we will discuss in Section 6.6.2.

6.3.4 Exploring the Effect of Rationales on User Decision-Making

We conducted an exploratory analysis to determine to what extent the attributes of rationale texts and UIs can influence users' decisions to grant, deny, dismiss, or ignore browser permission prompts. Additionally, we sought to evaluate how users perceive their overall experience with permission requests, particularly in terms of annoyance and ease of use. To that end, we relied on two datasets that we gained access to from Google's Chrome browser:

Chrome telemetry data: We analyzed Chrome desktop data on user interactions with permission prompts. This data is collected when users (1) enable the setting to "Make searches and browsing better" by sending URLs of visited pages to Google and (2) when at least 50 Chrome users visit the page and respond to a permission prompt. The data we used covers the 28 days leading up to August 8, 2024.

Chrome user sentiment data: Chrome fielded experience sampling questionnaires to understand how users feel about web permissions on desktop platforms. Users were eligible to answer a questionnaire if they had enabled the "Help improve Chrome's features and performance" setting and had not seen another questionnaire in the last 180 days. We focused on responses from users who shared URLs with their answers, which allowed us to link their feedback to specific reasons and user interface designs. URLs are available when users opted into the "Make searches and browsing better" setting. Questionnaires were available to Chrome users with an English language setting between November 2, 2023, and January 15, 2024, and collected 118,949 complete responses. The questionnaire was originally fielded for a Google-internal project and included four questions, two of which we were able to leverage to understand user sentiment on permission prompts. Respondents rated both their annoyance and ease of decision-making on a 5-point Likert scale (see Appendix 10.3 for exact wording and more details on the data collection).

We cross-referenced webpages with coded rationale texts and UIs with those containing telemetry and user sentiment data to obtain our sample described below.

Sample description. The telemetry sample included data for 242 of the websites with coded UI rationales and for 2,687 websites with coded text rationales. The user sentiment sample consisted of 1,351 responses across 282 URLs (169 geolocation, 59 camera, 55 microphone) for coded text rationales and 443 responses across 97 URLs (72 geolocation permission, 17 camera, 8 microphone) for coded UI rationales. Given that

Table 6.2: Summary of collected webpages and observed prompts.

	# Sites	# Pages	# Calls
Seed URLs	118,371	770,349	-
Collected successfully	113,537	739,235	-
Geolocation	22,036	99,241	1,608,729
Notification	6,657	69,567	73,231
Microphone	139	233	334
Camera	188	220	322
Total	29,020	**161,775**	1,682,616

these datasets reflect website usage, popular sites are more likely to appear, increasing the likelihood of detecting patterns common on more popular sites.

Additionally, we incorporated control samples from pages with permission prompts but no detected rationale for which we also had telemetry and user sentiment data available. For the UIs, we included all samples that were manually inspected but did not show a rationale for a total of 89 URLs. For the texts, we randomly selected 500 URLs from the set of manually verified text samples.

Statistical tests. We used regression models to evaluate the impact of rationale attributes on user behavior and sentiment toward permission prompts. Analyses were conducted in R 4.4.1. For user behavior, we applied exploratory linear regression to websites with telemetry data, modeling each prompt action (allow, deny, dismiss, ignore) separately. For sentiment, we used logistic regression on top-2-box Likert-scale scores. Permission type was included as a factor, acknowledging varying grant rates [71]. These models were exploratory and not further optimized or validated.

The following sections will provide a detailed presentation of the results for each section discussed above.

6.4 Web Crawling & Prompt Detection Results

In February 2023, we used the 770K seed URLs from Chrome telemetry dataset as a starting point to initialize our crawling infrastructure, deploying 100 parallel browser instances. As a result, we successfully collected snapshots of 739K pages from an EU vantage point. To ensure comprehensive coverage, we attempted to recrawl each failed page up to three times, followed by a manual review. For 31,114 URLs across 4,834 domains, all three crawl attempts failed–mostly due to inactive URLs or pages timing out (taking over 30 seconds to load). The data collection process spanned approximately seven weeks.

Table 6.2 provides an overview of the captured permission prompts and the triggered calls to permission-gated APIs recorded by our interactive crawler. It also quantifies the proportion of permission prompts observed within the Chrome telemetry data. Overall, our crawler found 29K domains with at least one web permission API call, with a

Table 6.3: Processing steps of the ∼6M unique text snippets extracted from in-the-wild webpages: (1) LLM filtering samples, (2) Large-scale BERT classification results, (3) Manually confirmed rationales, (4) Total count of rationale instances (real-world distribution), (5) unique webpages, and (6) unique domains.

	Processing		**Rationales**			
Permission	*LLM Filtering*	*Classifier*	*Confirmed*	*Instances*	*Pages*	*Sites*
Notification	6,918	2,675	1,666	22,739	14,855	1,950
Geolocation	127,552	2,305	1,063	2,136	1,680	894
Camera	14,082	1,005	495	848	543	364
Microphone	7,878	1,543	617	1,087	587	322
Total	155,093	7,254	**3,674**	**26,810**	17,333	3,237

total of ∼1.6M API calls across 161K webpages. We observed that the geolocation API is the most widely used, with almost 1.6M calls across 99K pages, and also the most widespread, being present on more than 22K sites, which is followed by push notifications present on 6.6K websites with 73K API calls.

Our crawler detected permission prompts on ∼20% of seed pages from Chrome telemetry, despite all using popular permission-gated web APIs (per Chrome telemetry). Investigating 100 missed cases revealed that 73% stemmed from crawling challenges like complex user interactions and authentication, 11% were inactive pages, and 16% involved privacy-sanitized URLs causing discrepancies. See Section 10.1.2 for further details. As we will show next, our approach can still detect rationales in many of these cases, particularly when the rationale text is present in the DOM, even if the permission prompt is not triggered by the crawler.

6.5 Rationale Identification Results

Starting with 739K pages across 113K websites from our seed list, we extracted ∼20M English text samples. After deduplication, we retained 6M unique samples.

6.5.1 Ground-Truth Dataset Creation with LLM Filtering

To create a relevant, smaller-scale dataset for training our ML classifier, we used few-shot prompting with the Mistral-7B LLM, as described in Section 6.3.2. We evaluated the LLM-based filtering on a manually compiled dataset of 143 rationales, selected via random sampling (denoted as DS1). The filtering approach achieved a high recall rate of approximately 93%, with 133 true positives (TPs) and 10 false negatives (FNs). Our goal was to maximize TPs while filtering out potential false positives (FPs), which would be handled later by the classifier. From the filtered set, we extracted about 155K unique rationale candidate texts from the 6M unique texts in our dataset. Table 6.3 presents an overview of these results and their distribution across permission types.

Next, we applied an iterative sampling method to the 155K filtered samples. We randomly selected batches of 100 samples for manual annotation until we gathered sufficient positive rationale samples. After reviewing 2,100 samples, we identified 262

rationales. Combined with the 143 rationales from DS1, this yielded 405 positive samples. We then added 1,785 negative samples to create a labeled training dataset of 2,190 samples, referred to as DS2.

We split DS2 into training, validation, and test sets using an 80%-10%-10% ratio. To address class imbalance, we under-sampled the training set to equalize the number of positive and negative samples. Using this labeled dataset, we trained a BERT classifier, as described in Section 6.3.2. The classifier achieved an F1 score of 0.82 on the validation set and 0.83 on the test set, indicating robust performance.

6.5.2 Large-Scale Rationale Classification

We used the BERT classifier to automatically annotate labels to the 155K samples and rule out potential false positives. The classifier flagged 7,254 unique samples as positive. These unique rationales correspond to 40,996 rationale instances across 28,538 unique pages that themselves belong to 5,798 unique domains.

6.5.3 Manual Review and False Positive Analysis

Three human analysts conducted a thorough manual review of all 7.2K discovered rationales to eliminate potential false positives following the methodology detailed in Section 6.3.2. When considering only the text, we observed a false positive rate of 19.6%, which is consistent with the figures observed in our test split during the training phase. When additionally considering the UI context, we observed a false positive rate of 49%, identifying 3,674 cases as true positives. The primary reason for this relatively high false positive rate is that many texts initially appeared to be valid rationales when evaluated in isolation. However, we observed that the context in which this text appears is crucial for accurate identification. For example, text snippets found on tutorial sites describing messages from other webpages, or user-generated content (e.g., comments) may initially seem like rationales but are not upon closer UI analysis. However, automatically extracting such contextual information from webpages remains highly challenging due to the dynamic nature of webpages and the complexity of HTML structures and semantics. This insight underscores the importance of both content and context in accurately identifying rationales on the web, which in this work, we tackled using a semi-automated approach.

6.5.4 Catalog of Rationales and Comparison with Permission Prompts

Through our extensive manual and automated analysis, we compiled a catalog of rationales containing 3,674 unique samples, totaling 26.8K instances across 17.3K unique webpages and 3,237 unique domains. Notification rationales were the most common, with over 22K instances, while camera rationales were the least common, with only 848 cases. A summary of our rationale catalog across various permission types can be found in Table 6.3 (columns 4-7).

Our crawler detected permission prompts on 161K webpages, but according to our ML-based detection, only 7.5K of those pages (4.6%) included a text-based rationale. This lower rate is expected, as many pages trigger prompts without including rationales,

Table 6.4: Rationale messages from libraries and their prevalence on the Web based on the number of webpages. The table shows the contribution of library code signature searching to identify additional webpages that use one of the rationale messages in our catalog, and compares it with results from our ML-based detection pipeline. The library detection rules are in Table 10.4. **Legend:** S = signature search. M = machine learning. $SITE= a placeholder for site name. N = Notification. G = Geolocation.

Rationale of Library		M ∪ S	M ∩ S	S only	M only
iZooto [82]:					
Real time notifications have been turned off. Enable them to get important and timely updates.	N	5,274	5,074	0	200
Real time notificatios are turned off. You can enable it to receive timely updates.	N	5,274	5,074	0	200
OneSignal [120]:					
We'd like to show you notifications for the latest news and updates.	N	64,194	473	63,716	5
Would you like to be aware of all the hottest news and events from $SITE?	N	55	1	55	0
PushEngage [129]:					
Subscribe to notification	N	730	205	525	0
Smart Push [78]:					
Give us a permission to receive push notification messages and we will keep you posted	N	1,174	652	522	0
Give us a permission to receive push notification messages and we will keep you posted	N	1,174	652	522	0
You can choose to turn off notifications later anytime using browser settings.	N	1,174	746	428	0
Moe-push [112]:					
This website would like to send you awesome updates and offers! Notifications can be turned off anytime from browser settings. Don't Allow	N	491	55	436	0
PushOWL [130]:					
Get Updated with Latest Offers and Products.	N	685	262	412	11
Perfecty [125]:					
Do you want to receive notifications?	N	354	90	264	0
I want to receive notifications	N	354	287	67	0
Webpushr [166]:					
You are unsubscribed to Push Notifications	N	397	192	201	4
You are subscribed to Push Notifications	N	397	202	191	4
Subscribe to receive push notifications on latest updates	N	400	207	186	7
You have blocked Push Notifications. Follow these instructions to enable Push Notifications.	N	397	208	185	4
Superstorefinder-wp [154]:					
Location service is not enabled. Continue anyway | Share my location	G	79	49	30	0
Storerocket [77]:					
Get notified of new locations.	N	56	41	14	1
Allow the geolocation on your browser and refresh the page.	G	75	71	3	1
Your browser blocked our request to get your location.	G	75	71	3	1
Total		**82,809**	14,612	**67,760**	438

or the rationales may be embedded in non-text formats. On the other hand, our ML-based approach identified 9.8K pages with text rationales where no prompt was observed, suggesting that rationales can still be present even when the crawler is unable to trigger a prompt.

6.5.5 Rationales in Libraries and Prevalence

We observed that certain rationales in our catalog exhibited a notably high prevalence across the web. Intrigued by this pattern, we manually reviewed the most frequent rationales. We found that these cases are associated with geolocation and notification permissions, and implemented via 10 distinct third-party libraries.

Library signatures. For each library, we extracted specific code signatures based on HTML elements (such as `id` and `name`) that these libraries incorporate into webpages. Our goal was to search these signatures within our broader dataset of webpage snapshots collected during web crawling, allowing us to uncover any instances that our machine learning-based detection pipeline might have overlooked. In total, we created 32 detection rules for libraries. The complete list of rules is in Table 10.4. As we show in Appendix 10.5, our rules are robust against false positives.

Libraries and prevalence. Table 6.4 presents the ten libraries we identified and their rationale messages. For each rationale, the table shows the number of webpages we found using our (i) machine learning-based approach and (ii) signature search approach, including the union and intersection of both methods. Our analysis reveals that signature searching significantly enhances our findings, uncovering over 67K additional instances of rationale messages for one of these libraries in the wild, compared to only 15K instances of these libraries detected using our initial dataset. We observed that the top three used libraries are OneSignal [120], iZooto [82] and Smart Push [78]. However, the majority of the newly discovered instances belong to OneSignal, with over 63K instances identified through signature searching alone. In addition, the majority of these 10 libraries are focused on push notifications, with only a few supporting or being designed for geolocation permissions. In total, we identify 82.8K webpages that use a rationale from a library, and 85,093 webpages that use either a custom or library rationale. Overall, this strategic approach not only refined our understanding of rationale prevalence but also strengthened the comprehensiveness of our rationale catalog. The ML-based approach missed these rationales because the captured webpage snapshots did not include the rationale text, which required complex user interactions (e.g., clicks) to appear. Our interactive agent in Section 6.3.1 failed to simulate these interactions. However, the HTML code signatures of libraries were present, enabling the signature-based method to find them. The ML-based approach was able to find the text of these rationales on other page snapshots that used the same libraries but did not require user interaction to load their content. We refer interested readers to Appendix 10.5, where we discuss the complementary nature of ML and signature-based rationale detection methods.

We note that when the library signatures appear in the DOM, we cannot guarantee that the rationale message will be always visible. Also, as we will discuss in Section 6.8.1,

Table 6.5: Overview of text-based rationale clusters ordered by size. The top part shows prevalent clusters, while the bottom part highlights unique behaviors that are grouped in the *other* cluster. The left part shows the percentage of *unique* samples per cluster, whereas the right part (i.e., instances) shows their real-world distribution, i.e., non-unique count of rationales across webpages in our data set.

Cluster	Subcl.	Rat.	Pct.	Inst.
C1: Notification	28	1,655	45%	22,727
C2: Geolocation	21	1,046	28.4%	2,116
C3: Microphone	11	468	12.7%	877
C4: Camera	7	348	9.4%	641
C5: Camera_Microphone	2	139	3.7%	197
C6: Other	6	18	0.49%	21
Notification_Geolocation	1	7	0.19%	7
Camera_Microphone_Geolocation	1	4	0.11%	5
Camera_Notifications_Microphone_Geolocation	1	3	0.08%	4
Microphone_Location	1	2	0.05%	3
Notifications_Microphone	1	1	0.03%	1
Camera_Geolocation	1	1	0.03%	1
Total	70	3,674	100.00%	26,810

it is challenging to fully disentangle library and custom rationales at scale, since webpages may use both or customize them.

6.6 Analysis of Rationale Text & UI Patterns

We analyzed the collected rationales to identify common patterns, following the methodology outlined in Section 6.3.3.

6.6.1 Rationale Text Patterns

Our automated clustering method leveraging all-MiniLM-L6-v2 sentence transformer organized the 3.6K rationales into 75 clusters based on both textual syntax and semantics. Out of these, we consolidated six clusters that included only few samples, into a larger one named *Other*, reducing the total to 70 clusters. To simplify, we applied hierarchical clustering and further merged clusters that rely on the same set of permissions together, resulting in six higher-level clusters. Table 6.5 provides a summary of the clusters by permission type, detailing both their unique count and their prevalence on the web. We refer interested readers to Appendix 10.6, which provides examples of the 70 subclusters.

We found that the *notification* cluster is the largest, making up about 45% of the unique samples with over 22K instances observed in the wild. The *geolocation* cluster follows, representing over 28% of unique rationales in our catalog, but with significantly lower prevalence at around 2.1K instances.

Rationale text attributes. We undertook a qualitative analysis of the 70 clusters to extract their characteristics by manually examining random samples from each cluster

Table 6.6: Distribution of attributes and corresponding action rates across webpages with available telemetry ($n = 2,675$). The *"None"* attributes serve as control groups. Attributes in Message Content are not mutually exclusive, we omit action rates there. The count column indicates the number of webpages with rationales containing each attribute (per Section 6.3.3) for which telemetry was available. Percentages in brackets show the proportion of webpages exhibiting the attribute.

			p50 (Median)			
Category	**Attribute**	**Count**	**grant**	**deny**	**dismiss**	**ignore**
Tone	None	498 (18.6%)	15.1%	7.8%	31.0%	35.3%
	Neutral	2,026 (75.5%)	12.5%	8.9%	25.5%	41.2%
	Negative	102 (3.8%)	48.2%	6.2%	20.3%	11.6%
	Positive	49 (1.8%)	39.5%	6.3%	24.5%	21.2%
Encourage	None	2,386 (89.2%)	14.5%	8.4%	26.5%	37.8%
	Motivation	284 (10.6%)	12.1%	11.3%	24.1%	45.3%
	Consequence	5 (0.2%)	60.7%	3.3%	32.1%	1.6%
Required Perm.	True	66 (2.5%)	66.6%	5.5%	17.9%	6.1%
	False	2,609 (97.5%)	13.5%	8.7%	26.4%	39.5%
Optional Perm.	True	23 (0.9%)	53.2%	5.4%	22.8%	10.4%
	False	2,652 (99.1%)	13.9%	8.7%	26.3%	39.0%
Msg. Content	Perm. Request	1,361 (50.9%)	18.7%	7.8%	24.5%	31.7%
	Func. Expl.	311 (11.6%)	18.6%	9.8%	19.7%	31.5%
	Error	265 (9.9%)	43.1%	6.3%	20.3%	14.4%
	Instruction	207 (7.7%)	8.7%	11.1%	23.6%	50.4%
	Emphasize Control	85 (3.2%)	14.6%	9.0%	26.1%	39.6%
	Data Use Reassur.	44 (1.6%)	67.2%	5.1%	17.2%	5.8%
	Loading Device	14 (0.5%)	91.9%	2.2%	2.9%	1.5%

until saturation, in line with the methodology detailed in Section 6.3.3. Other than permission type, we found that rationale texts can vary widely across four dimensions: (i) sentiment and tone, (ii) encouragement style including benefits and consequences, (iii) necessity of permission granting for proper functionality, (iv) and message content such as errors, instructions, and reassurance on data use. In the following, we discuss these attributes with real-world examples.

Message Tone: The tone of a rationale text indicates the emotional or attitudinal stance conveyed to the user. A *positive* tone (POS) employs language that contains excitement, such as the message *"This website would like to send you awesome updates and offers!"* In contrast, a *neutral* tone (NEUT) presents information in a factual manner, as seen in examples like *"This website requests access to your location."* Conversely, a *negative* tone (NEG) communicates caution or potential errors, as illustrated by statements like *"Sorry! We can't access your webcam and/or audio recorder."*

Encouragement: Encouragement in rationale texts could vary. A *motivating* approach (MOTIV) suggests actions by highlighting benefits, exemplified by statements like *"Allowing notifications will keep you updated with the latest news"* or *"Granting camera access will improve your experience."* Instead of the benefits, the message may

convey the *consequences* of permission denial (CONSEQ), e.g., *"WARNING: If you select BLOCK, you cannot have a video call because your camera and microphone cannot be used"* or *"Blocking camera access may limit features of this website"*.

Permission Necessity: Necessity in rationales determines the perceived importance of a permission request. *Required* actions (REQU) indicate essential permissions, e.g., *"Permission to access your contacts is required to sync data."* These cases often directly instruct or mandate user action, such as stating, *"You must grant microphone access to continue."* Conversely, *optional* actions (OPT) suggest enhancements or alternatives, such as *"You can enter your address manually or allow automatic filling for convenience"* and *"You may get a popup asking you to Allow or Block your location. The search function will work with either option."*

Message Content: Rationale messages include different and sometimes multiple types of content. One type provides *guidance* for troubleshooting and resolving problems, such as *"Troubleshoot permission issues by resetting your browser settings"*. These types of rationales may also provide more precise, possibly step-by-step instructions (INSTRUCT), such as *"To enable microphone access, go to Settings > Privacy > Microphone."* In comparison, *error* messages (ERROR) alert users about incorrect actions or issues, as seen in messages like, *"Error: Location access denied. Please grant permission to proceed."* Other *Emphasize control* (CONTROL), such as *"Notifications can be turned off anytime from browser settings."* In addition, the rationale can *reassure* users about data usage and privacy risks (REASSURE), e.g., *"Access to your camera is necessary, but no personal data is collected"* and *"We need your location to provide you with the best experience. Your location is safe with us."* Other rationales state the permission status, such as *"Accessing camera, please wait..."* and *"Waiting for camera to load"* (LOADING), or simply contain a direct *permission request* (PREQ) like *"Please allow access to your location."* Finally, the message may also include a functionality *explanation* (FUNC_EXPL) clarifying for what purpose the permission is needed, e.g., *"In order to find a store near you, allow location access."* Websites rely on these types of rationales based on their specific scenarios and user needs.

6.6.1.1 Analysis of Rationale Text Attributes

Table 6.6 shows the prevalence of each rationale attribute across the webpages in our dataset for which we had telemetry available (overall $n = 2,675$) and the corresponding permission prompt action rates. Our clustering algorithm of Section 6.3.3 identified 123 rationale groups across 17.3K URLs. However, only 32 groups had samples from more than 10 URLs, and among those, only 18 had sufficient telemetry data. We focused on these 18 groups to analyze how various factors, such as message tone and encouragement, influence users' permission decisions.

Common text patterns. Table 6.7 provides an overview of the 18 common rationale clusters and their sizes, illustrating how various online platforms communicate their need for web permissions. Each cluster is categorized by a distinct set of attributes

Table 6.7: Summary of text rationale clusters having more than 10 samples based on their distinct number of URL-permission pairs for which Chrome telemetry was available. For each cluster, the table shows its attributes, unique count (i.e., number of URLs in the dataset belonging to that cluster), number of cluster URLs with telemetry data, and an example. **Legend:** Cl. = Cluster. $SITE = a placeholder for site name. N = Notification. G = Geolocation. C = Camera. M = Microphone.

Cl.	Attributes	Rationale Example	Count	Telem.
N0	NEUT	Click Allow to receive notifications.	11,863	882
N4	NEUT, PREQ, FUNC_EXPL	click Allow to get notified about low-cost dental care	663	171
N33	NEUT, INSTRUCT	Step 2. Tap the toggle switch to turn the notification off and on.	470	147
N2	NEUT, CONTROL	The website $SITE would like to send you push notifications. Notifications can be turned off anytime from browser settings.	1,031	46
N20	POS, MOTIV, PREQ, CONTROL	$SITE would like to send awesome offers for your furry friend! Notifications can be turned off anytime from browser settings. Don't Allow	247	26
N14	NEUT, INSTRUCT, PREQ	Subscribe to receive push notifications on latest updates You have blocked Push Notifications. Follow these instructions to enable Push Notifications.	239	15
N35	NEUT, MOTIV, FUNC_EXPL	Apply to jobs anytime, anywhere and get notified instantly when your application is reviewed.	32	13
G1	NEUT, PREQ, FUNC_EXPL	Please allow location permission from your browser to view nearby leases	1,233	432
G10	NEG, ERROR	Opps! Unable to retrieve your location, please enable location access in your browser OK No Cancel	198	62
G29	POS, MOTIV, PREQ	$SITE requires access to location. To enjoy all that $SITE has to offer, turn on your GPS and give $SITE access to your location.	47	13
G34	NEUT, ERROR, REASSURE	Your Location access is blocked! Please provide location access to proceed further. Your location is safe with us.	22	12
C6	NEUT	You will be asked to enable camera access	248	64
C17	NEUT, PREQ	When prompted, click "Allow" and you'll see your camera	50	28
C9	NEUT, INSTRUCT	Use your Camera to start VideoChat. Allow access the Camera in your browser's Settings Your webcam is active on $SITE. To use the webcam here please close $SITE	36	10
M7	NEUT, PREQ	Enable microphone access on this site by clicking the big "Enable Microphone" button.	163	56
M5	NEUT	Use the audio devices on your computer to speak and listen	101	40
M27	NEUT, FUNC_EXPL	After pressing the call button, a window appears in the upper right corner asking you to allow access to the microphone, you should click enable to start the free call. If you accidentally clicked on the disallow button, try reloading the page.	70	29
M96	NEG, ERROR, FUNC_EXPL, REQU	No microphone found. Unable to continue.	13	13

from Section 6.6.1, capturing their tone, encouragement, necessity, and message content. The N0 cluster is the most prevalent, appearing on over 11,8K webpages. These neutral messages typically prompt users to allow notifications and have significant telemetry presence, with 882 URLs, showing that users frequently encounter this scenario online. Another key cluster, G1, appears on over 1.2K webpages and underscores the widespread use of geolocation rationales. Lastly, the N2 cluster for push notifications, appears on nearly 1K URLs and reassures users that they can deactivate notifications at any time. However, the lower telemetry counts suggest these pages are less frequently visited.

Message Tone: We observed that the majority of the rationale messages (75.5%) maintain a neutral tone across most clusters, such as N0, G1, C6 and M7 in Table 6.7, aiming for a straightforward and factual manner. Positive and negative tones are rarer, accounting for about 2-4% of our samples. Positive tones are employed mostly in notification and geolocation rationales such as N20 and G29, together with motivations that highlight the benefits of granting permissions, such as receiving offers or enhancing their experience. Negative tones are often used in geolocation and microphone rationales, such as G10 and M96.

Encouragement: Encouragement strategies that highlight the benefits of granting permissions, such as receiving timely updates, discounts, and price alerts via push notifications, or ensuring the best experience through location tracking, were common, appearing in 10.6% of samples, as seen in clusters like N20 and G29. In contrast, using consequences as a cautionary tactic is notably rarer, appearing in just 0.2% of our dataset, typically to warn users about potential limitations in functionality if permissions are not allowed.

Permission Necessity: We found that most rationales avoid explicit categorization, with only 2.5% labeling actions as required and 0.9% as optional. Instead, the required permissions are only clearly emphasized in critical contexts where the service cannot function without specific permissions, such as the mandatory geolocation permission for account creation (geo-restricted) in cluster G0 of Table 10.5 and M96 for microphone access in Table 6.7.

Message Content: Rationale messages vary in content, often combining multiple types. The most common are permission requests (50.9%), such as G1, N4 and M7, mirroring the primary purposes of most rationale texts. Other notable types include emphasizing that the permission can be deactivated anytime (3.2%), such as N2, and the provision of functionality explanations (11.6%), such as M27. These elements highlight the efforts of websites to inform users and reinforce control [20]. Rationales involving error messages (9.9%) and guiding instructions (7.7%) are less frequent but commonly used in scenarios where user action is needed to resolve issues, such as N33, C9, G34, and M96. Finally, reassurances on data use are rare (1.6%), like G34.

Summary of Insights. Overall, our systematic analysis reveals a spectrum of strategies in rationales, reflecting varying levels of urgency and user autonomy. The distinctions across clusters underscore the tailored approaches websites take depending on the specific functionality and sensitivity of the data involved, showing the intricate balance between user experience and operational necessity. The distribution of the identified rationales highlights a preference for neutral messaging, with larger clusters reflecting more common and broadly applicable requests. These messages generally avoid explicitly stating the necessity of permissions, allowing users to infer their importance. In contrast, the smaller clusters, while less frequent, are mostly tailored to specific user interactions, such as permission-dependent functionalities (e.g., camera for identity verification), troubleshooting or reassurance, which may require more detailed or emotionally nuanced messaging.

6.6.2 Rationale UI Patterns

We analyzed rationale UIs following the methodology described in Section 6.3.3, compiling a dataset of 749 rationale designs across 631 webpages. Our analysis revealed several patterns in how permission requirements are communicated, identifying eight distinct layout patterns. Each layout is characterized by a unique combination of attributes, which we will introduce first in the following section.

Rationale UI attributes. A rationale, which may be linked to one or more permissions, has a distinct layout with several notable attributes, which we identified as follows:

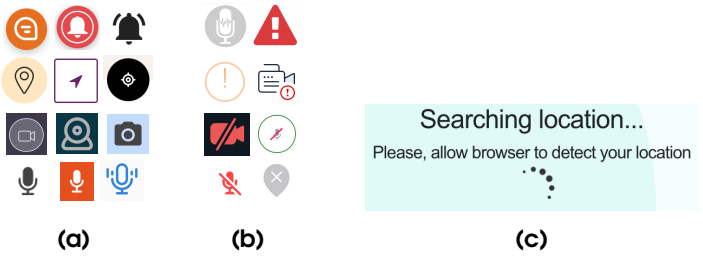


Figure 6.2: Icons in rationales. (a) Representing permissions, from top to bottom: notifications, geolocation, camera, and microphone. (b) Indicating denied permissions. (c) Loading icon in a rationale displayed alongside a browser prompt.

Position: Rationales can be *inline*, seamlessly integrated into the page, or *floating*, overlaying content or protected elements like a camera feed or map. Floating rationales can appear anywhere on a 2D plane: top, bottom, left, right, or center.

Elements: Rationales often include additional elements such as *buttons*, *icons*, *alternative options*, and *visual or textual instructions*. Regarding *buttons*, we identified three types based on their functionality. A positive button is designed to trigger the browser prompt when clicked, typically labeled with *"Allow"*, *"Subscribe"*, *"Use My Location"*, or *"Turn On"*. Conversely, acknowledge/dismiss buttons serve to acknowledge

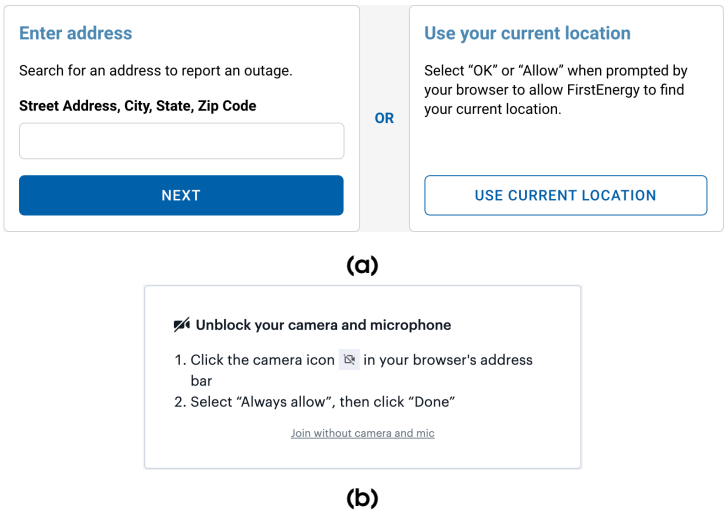


Figure 6.3: Buttons in rationales. **(a)** Option to access their current location or manually search for an address. **(b)** Option to join a meeting without granting permissions.

and dismiss the rationale and were labeled with *"OK"*, *"Got It"*, *"Cancel"*, *"Don't Allow"*, or *"Later"*. Websites represented these buttons also by an *"X"*, typically located in the upper right corner of the rationale. We also found instances where links were provided to help and troubleshooting pages, labeled with phrases such as *"How to Grant Access"*, *"Troubleshooting Tips"*, or *"The Help Center"*. We found that when multiple buttons are present, developers often use opinionated design by applying distinct styling–such as different colors, sizes, or visual cues–to the positive button. While this approach can encourage users to grant permissions, not all patterns designed to increase grant rates are legitimate. In some cases, such nudging crosses into dark patterns that manipulate users into granting permissions they might not otherwise allow. We leave investigating their prevalence for future work.

Besides buttons, we observed that rationales may include *icons* referencing the protected functionality, as illustrated in Figure 6.2a. Icons may also indicate permission status, such as a crossed-out or disabled icon for denied permissions, or an exclamation mark signaling missing permissions, as shown in Figure 6.2b. Then, instead of granting permission, rationales may offer *alternative options* to users. For example, users might manually search for a location instead of granting geolocation access (Figure 6.3a), join an online meeting without camera and microphone access (Figure 6.3b), or view all shops instead of only seeing the nearest ones. Finally, rationales can include *visual or textual instructions*, using screenshots (Figures 6.4a, 6.4c) or text (Figure 6.4b).

Timing: Rationales can be introduced at various points of the permission request cycle. *Before* requesting permission, a rationale can prepare the user for the request. *Alongside* a browser prompt, a rationale can guide the user's attention, often using a loading icon to indicate the webpage is waiting, such as Figure 6.2c. *After* dismissing, ignoring, or denying, a rationale can inform the user about the missing permission using indicators like bright colors, bold typography, and warning icons, as in Figure 6.2b.

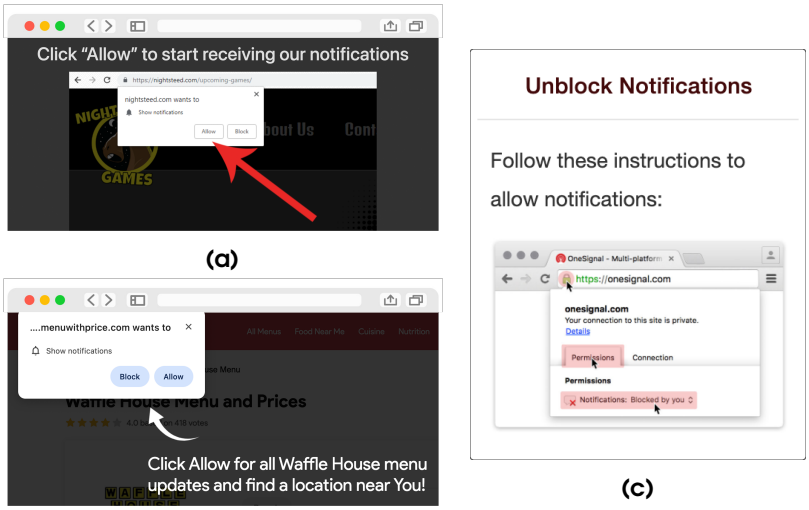
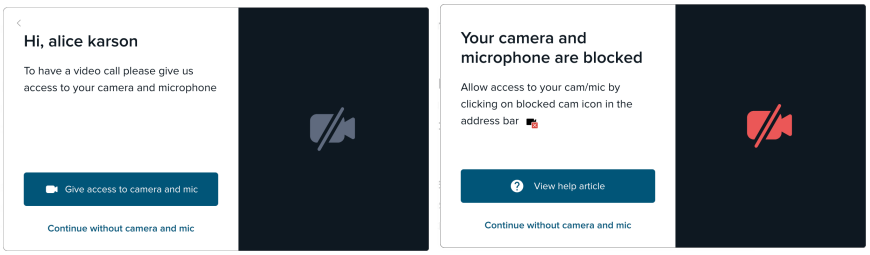
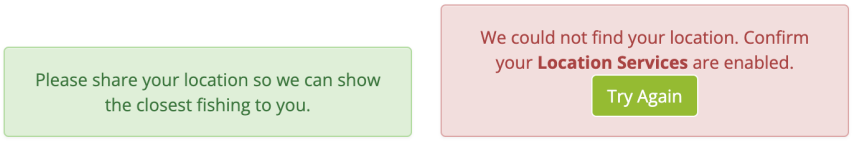


Figure 6.4: Textual and visual instructions. (a) As screenshot showing how to grant permission when prompted. (b) With an arrow directing attention to the browser prompt. (c) Showing how to re-enable denied permission.



(a) Before browser prompt. **(b)** After permission is not granted.



(c) Before browser prompt. **(d)** After permission is not granted.

Figure 6.5: Adaptive rationale with content updates after permission is not granted, shown as a dialog in (a) and (b), and as a banner in (c) and (d).

A webpage may present different rationales for the same permission depending on the timing. Some rationales remain static regardless of permission status, while others update dynamically. For example, an initial dialog may update with instructions on how to re-enable permission if denied, as shown in Figures 6.5a and 6.5b. Similarly, a banner rationale might change its content and color after permission is not granted (Figures 6.5c and 6.5d), and a fullscreen rationale can also differ before and after, as can be seen when comparing Figure 6.12a with Figure 6.12b.

6.6.2.1 Analysis of Rationale UI Patterns

We outline the most common UI patterns, organized by layout and the frequently co-occurring attributes, providing a foundation for our exploratory analysis.

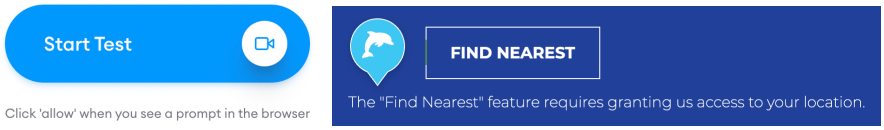


Figure 6.6: Buttons implicitly requesting permission, with explanatory text.

Text: Starting with the simplest pattern, rationales can be solely text integrated within static webpage content. These rationales remain unchanged regardless of permission status. For instance, a webpage might state: *To use live audio input, please allow access to your browser microphone when prompted or check your browser settings."* Similarly, a help section might include steps like: *1) Plug in your headphones. 2) Allow browser access to your microphone. 3)..."*

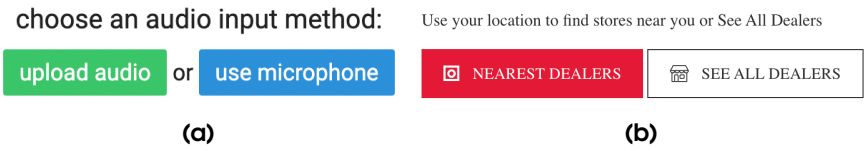


Figure 6.7: Rationale as button with alternative option.

Button: Buttons, such as dismiss or allow buttons on a dialog, can be part of a larger rationale layout. However, buttons can also serve as the rationale itself, implicitly indicating the need for permission to enable a feature. Examples include buttons labeled *"Find Immediate Care Near You"* or *"Start Video Chat."* Explanations may accompany buttons, as in Figure 6.6. For instance, a button labeled *"Start Test"* might be accompanied by the text *"Click 'allow' when you see a prompt in the browser."*

Whenever a button implicitly requests permission by activating a function or explicitly with labels like *"Allow Permission"*, an alternative to granting permission can be provided. For example, instead of granting microphone access, the user might click a button to upload an audio file (Figure 6.7a). Similarly, instead of clicking on *"Places Nearby"* button, the user could also use the *"See All Places"* option (Figure 6.7b).

Particularly for notification permissions, clicking a button can trigger the display of a rationale message. We observed several variations in presenting this rationale. For instance, clicking a button with a notification-related icon, such as a bell, may display a rationale dialog either in the center of the screen or as a floating element above the button, as depicted in Figure 6.8. Alternatively, the button may expand horizontally to form a banner that contains the rationale message, as in Figure 6.9.

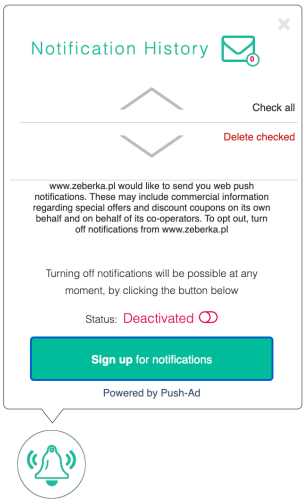


Figure 6.8: Rationale dialog appears after clicking on bell button.

Banner: Rationales in the form of a banner appear dynamically when awaiting user interaction with a permission prompt (see Figure 6.10a) or after permission is not granted (Figure 6.10b). In the latter case, they are usually displayed on a prominently colored banner, often red, featuring exclamation mark icons, distinct typography, and variably colored text. Banners are often displayed inline with the webpage content but can also float above it, spanning the full width of the screen, mostly at the top as a header and occasionally at the bottom as a footer.

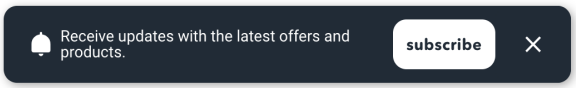
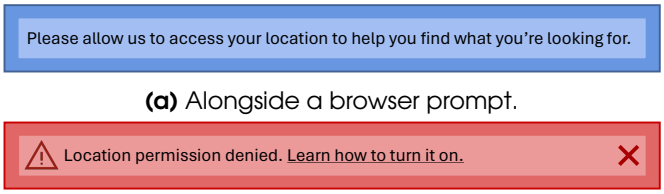


Figure 6.9: Rationale button expands to a banner after click.

Overlay: Rationales in the form of overlays appear on top of the main content of a webpage, often dimming or obscuring the background to draw the user's focus to the rationale. These overlays frequently guide the user to the browser prompt with an arrow, as depicted in Figure 6.11a. While most overlays can be dismissed with an *"X"* button, we also observed instances where the overlay remains persistent until the user interacts with the browser prompt or re-enables a previously blocked permission. For the 'quieter' notification permission [72], a rationale overlay can point to the address bar, directing the user on how to enable notifications.

Fullscreen: When a fullscreen rationale is displayed alongside a browser prompt, it operates like an overlay rationale but with a solid background. This rationale prompts the user to click *"Allow"*. Similar to overlays, the user must take action to proceed since the current only contains the rationale, as illustrated in Figure 6.11b.



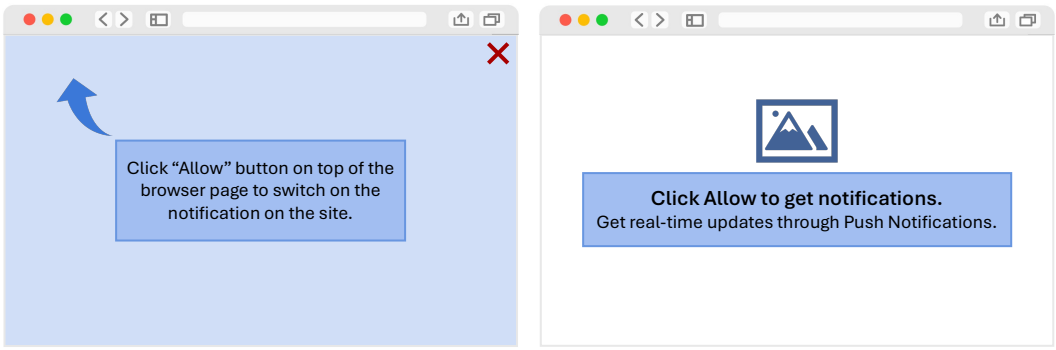
(a) Alongside a browser prompt.

(b) After permission is not granted.

Figure 6.10: Rationales as banners.

When the fullscreen rationale is shown before a browser prompt, it is typically part of a multi-step process to access a specific functionality. Users first grant permission, after which they can use the permission-protected feature. For instance, the fullscreen rationale in Figure 6.12a prepares the user for an upcoming browser prompt that appears when they click the button labeled *"Next,"* *"Enable Location,"* or *"Get Started"*.

If permission is not granted, the rationale prompts the user to grant permission and try again. Similar to rationales shown post-non-granting, it may include a *"Try Again"* button or a link to a help page. Instructions, often in the form of steps or screenshots, guide users on how to grant the necessary permission (Figure 6.12b).



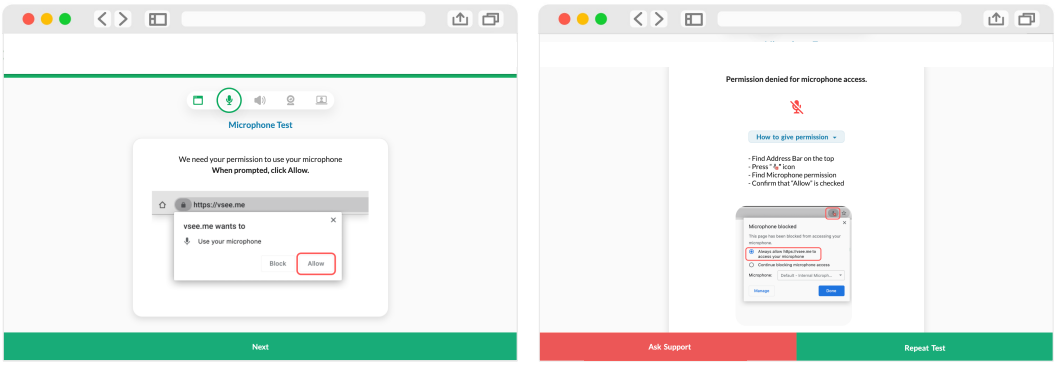
(a) Overlay.

(b) Fullscreen.

Figure 6.11: Rationales alongside a browser prompt.

Dialog: Dialog rationales are typically centered on the screen, either in the middle or, particularly for notification permissions, at the top. Dialogs overlay the webpage content, which can be darkened to provide emphasis. Figure 6.13 shows a common notification permission dialog.

Dialogs presented before the browser prompt generally have an explicit button for granting permission, such as *"Allow Location"*, *"Detect My Location"*, or *"Give Access to Camera and Mic"*. In addition, they may include a dismiss button, such as *"OK"*, *"Got It,"* or an *"X"* button. Figure 6.14a shows a rationale dialog before the browser prompt. We observed that dialogs are also used to inform users when permission is required but has been denied. In these cases, dialogs may feature an icon indicating that permission is missing, as shown in Figure 6.14b.



(a) Before browser prompt. **(b)** After permission is not granted.

Figure 6.12: Rationales as fullscreen.

On Permission-Protected Content: In previous patterns, we found that rationales can appear as floating elements on webpages, such as banners or dialogs. Additionally, they can overlay or replace permission-protected content, especially for geolocation and camera permissions, covering maps or camera/video feeds. When displayed before or alongside a browser prompt, these rationales ask the user to grant the necessary permission, as illustrated in Figures 6.15a and 6.15b. Conversely, Figures 6.15c and 6.15d show a rationale over protected content when permission is not granted.

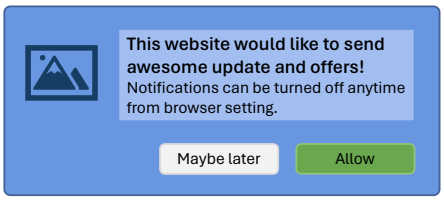
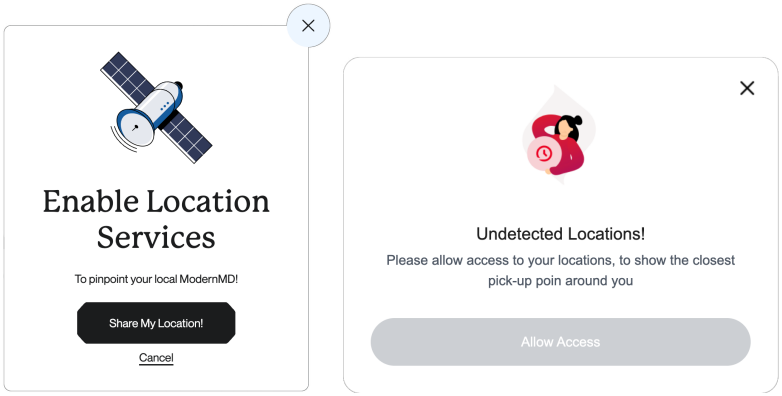


Figure 6.13: Example of a notification rationale dialog displayed at the top center of the screen before browser prompt.

Side of Map: In the context of geolocation permission, a rationale is displayed on the sidebar of a map. Typically positioned on the right-hand side, this sidebar is utilized to present a list of nearby places, a feature that becomes inaccessible when the permission is denied. In such cases, the rationale message informs the user that *"current location could not be determined."* However, users are often provided with alternatives. They may be prompted to explore all locations by clicking a button labeled *"Show All Locations,"* or they can manually search for a specific location using the search bar.

Common UI patterns. Table 6.8 shows the distribution of rationale layouts both before or alongside a browser prompt and following not granting permission. In total, we clustered 387 rationales for geolocation, 149 for notification, 88 for camera, 53 for microphone, and 72 for both camera and microphone permissions across the eight distinct layout patterns described above. We found that geolocation rationale UIs often



(a) Before browser prompt. **(b)** After permission is not granted.

Figure 6.14: Rationale as dialogs.

include an alternative option with common layouts such as buttons, dialogs, text, and side-of-map rationales. Notification rationales typically appeared before or alongside a browser prompt, primarily as dialogs or fullscreens without alternatives. Rationales for camera and/or microphone were presented as text, followed by dialogs and fullscreens. For Microphone permissions, text and buttons were the most frequent layouts.

Summary of Insights. Our analysis of rationale UIs reveals eight common layout patterns: *text*, *buttons*, *banners*, *overlays*, *fullscreens*, *dialogs*, *on-permission-protected content*, and *side-of-map*. Each pattern serves distinct purposes, such as presenting static or dynamic messages (text), encouraging action (buttons), or offering alternatives (e.g., for geolocation or camera permissions). Patterns like banners, dialogs, and overlays are often used dynamically to emphasize missing permissions, while fullscreens guide user interactions through focused layouts. Different permissions, such as geolocation, notification, and camera, influence the choice and frequency of these patterns.

6.7 Exploring the Effect of Rationales on User Decision-Making

This section presents an overview of our exploratory analysis of how text attributes and UI patterns in rationales influence user actions on permission prompts. As detailed in Section 6.3.4, we use the action rates from Chrome telemetry and user sentiment data from Chrome experience sampling.

6.7.1 Analysis of Rationale Text Attributes

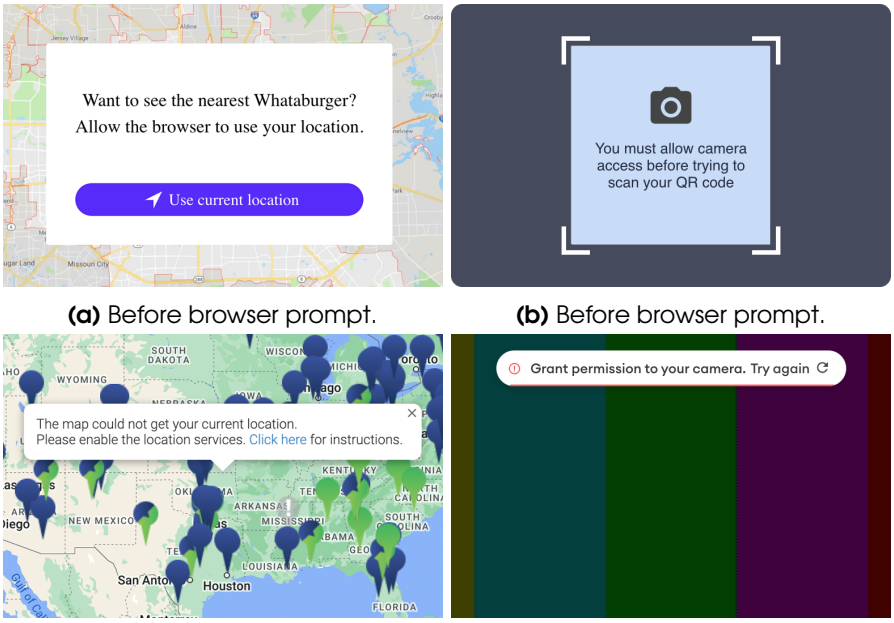
Table 6.6 provides an overview of the action rates across the various features of rationale texts we identified and Table 6.9 describes the regression models we fitted to explore the effects for each of the attributes we identified.

Table 6.8: Overview of most common rationale layouts per permission type.

	Geolocation		**Notification**		**Camera**		**Microphone**		**Cam & Mic**	
	Before	*After*	*Before*	*After*	*Before*	*After*	*Before*	*After*	*Before*	*After*
Count	**183**	**204**	**119**	**30**	**51**	**37**	**33**	**20**	**35**	**37**
Percent	47%	53%	80%	20%	58%	42%	62%	38%	49%	51%
Text	8.8%	13.7%	7.4%	–	35.2%	11.4%	30.2%	5.7%	12.5%	<5.0%
Button	18.9%	<5.0%	<5.0%	–	8.0%	<5.0%	17%	7.5%	9.7%	–
Banner	<5.0%	12.7%	<5.0%	<5.0%	<5.0%	5.7%	<5.0%	9.4%	–	6.9%
Dialog	10.6%	14.2%	42.9%	8.7%	<5.0%	10.2%	<5.0%	<5.0%	9.7%	18.1%
Fullscreen	<5.0%	<5.0%	21.5%	<5.0%	8.0%	9.1%	11.3%	11.3%	9.7%	13.9%
Overlay	<5.0%	<5.0%	<5.0%	6.7%	<5.0%	<5.0%	–	–	5.6%	<5.0%
OnProtect	<5.0%	<5.0%	–	–	<5.0%	<5.0%	–	<5.0%	<5.0%	6.9%
SideMap	–	6.2%	–	–	–	–	–	–	–	–
Alternative	36.4%		–		<5.0%		<5.0%		<5.0%	

Table 6.9: Results of four exploratory regression models using the four user action rates in permission prompts as dependent variables. In dependent variables include the permission type as well as the identified rationale text attributes. Reference categories are geolocation for permission type, as well as "none" (i.e., no rationale text being present at all) for tone and encouragement. The remaining factors are binary, i.e. encode whether or not the given content type was present or not. **Legend:** SE = Standardized Error. * p <.05, ** p <.001, *** p <.0001.

	Grant Rate *Estimate (SE)*	**Deny Rate** *Estimate (SE)*	**Dismiss Rate** *Estimate (SE)*	**Ignore Rate** *Estimate (SE)*
Intercept	0.25 (0.009)***	0.10 (0.003)***	0.36 (0.006)***	0.28 (0.009)***
Permission				
Camera	0.37 (0.014)***	-0.06 (0.005)***	-0.16 (0.010)***	-0.15 (0.015)***
Microphone	0.43 (0.013)***	-0.07 (0.004)***	-0.20 (0.010)***	-0.16 (0.014)***
Notification	-0.24 (0.008)***	0.00 (0.003)	-0.06 (0.006)***	0.30 (0.009)***
Tone				
Negative	0.05 (0.023)*	0.00 (0.009)	-0.05 (0.017)**	0.00 (0.024)
Neutral	0.08 (0.010)***	0.02 (0.003)**	-0.05 (0.008)***	-0.04 (0.011)***
Positive	0.18 (0.032)***	-0.02 (0.011)	-0.06 (0.023)*	-0.11 (0.033)**
Encouragement				
Motivation	-0.03 (0.013)	0.02 (0.005)***	0.00 (0.010)	0.01 (0.014)
Consequences	-0.20 (0.078)**	0.01 (0.026)	0.12 (0.056)*	0.07 (0.080)
Functionality				
Required	0.08 (0.022)***	-0.01 (0.007)	-0.03 (0.016)	-0.05 (0.023)*
Optional	-0.28 (0.037)	0.01 (0.012)	-0.02 (0.030)	0.03 (0.038)
Message Content				
Error	0.03 (0.014)	0.00 (0.005)	-0.01 (0.01)	-0.01 (0.015)
Instruction	0.00 (0.013)	0.01 (0.004)***	-0.01 (0.009)	0.00 (0.013)
Func. Explanation	0.04 (0.013)**	0.02 (0.004)***	-0.01 (0.009)	-0.04 (0.013)***
Loading Device	0.11 (0.047)*	0.01 (0.015)	-0.06 (0.034)	-0.07 (0.048)
Data Use Reassurance	0.11 (0.028)***	-0.02 (0.009)*	-0.06 (0.020)	-0.03 (0.028)
Permission Request	0.30 (0.008)***	-0.02 (0.003)***	0.00 (0.006)	-0.01 (0.008)
Emphasize Control	0.06 (0.021)**	0.01 (0.007)	0.00 (0.015)	-0.08 (0.022)***



(a) Before browser prompt. **(b)** Before browser prompt.

(c) After permission is not granted. **(d)** After permission is not granted.

Figure 6.15: Rationales on permission-protected content. (a) and (c) show permission on map. (b) and (d) show permission on camera feed.

6.7.1.1 The Effect of Text Attributes on Permission Prompt Actions.

Our analysis suggests user behavior on permission prompts is influenced by how rationale messages are composed, primarily on grant rates. Following, we detail the effects.

Insight #1: Message Tone. First and foremost, having any rationale message, even with a neutral or negative tone, positively influences the grant rate and is associated with lower dismiss and deny rates. Neutral and in particular positive tones in rationale texts are associated with higher grant rates (+8% and 18%, respectively).

Insight #2: Encouragement. The use of consequences as an encouragement strategy significantly decreases the grant rate (-20%). This attribute is also associated with a 12% higher dismiss rate. Highlighting benefits seems to have a small negative impact, with slightly lower grant rates (-3%) and slightly higher deny rates (+2%).

Insight #3: Permission Necessity. When functionality is stated as required in the rationale text, the grant rate is somewhat higher (+8%) and prompts get ignored slightly less frequently (-5%). Mentioning optional and alternative functionalities does not appear to have significant effects on user actions.

Insight #4: Message Content. Including additional content in rationale messages generally leads to higher grant rates (+3% to +30%). Among these, the most common and relatively neutral type had the strongest positive effect in our sample. Messages

Table 6.10: Results of six exploratory regression models with user action rates, user annoyance, and ease of decision-making as dependent variables. Independent variables are permission type and the identified rationale UI clusters. Reference categories are geolocation for permission type, and "none" (i.e., no rationale being present at all) for before browser prompt and after permission denial. The additional factors are binary, i.e. encode whether or not the given element was present or not. **Legend:** SE = Standardized Error. Odds R. = Odds Ratio. Not Annoy. = Not Annoying. Before = Before browser prompt. After = After permission denial. * p <.05, ** p <.001, *** p <.0001.

	Grant *Estimate (SE)*	**Deny** *Estimate (SE)*	**Dismiss** *Estimate (SE)*	**Ignore** *Estimate (SE)*	**Not Annoy.** *Odds R. (SE)*	**Is Easy** *Odds R. (SE)*
Intercept	0.32 (0.02)***	0.11 (0.01)***	0.32 (0.01)***	0.24 (0.02)***	6.89 (1.40)***	3.35 (1.31)***
Permission						
Camera	0.27 (0.04)***	-0.06 (0.01)***	-0.14 (0.03)***	-0.07 (0.04)	5.05 (2.39)	0.66 (1.65)
Mic.	0.26 (0.05)***	-0.06 (0.01)***	-0.16 (0.03)***	-0.04 (0.05)	2.20 (2.03)	0.97 (1.73)
Notif.	-0.30 (0.03)***	0.00 (0.01)	-0.01 (0.02)	0.31 (0.03)***	–	–
Before						
Banner	0.08 (0.12)	0.02 (0.03)	0.04 (0.07)	-0.14 (0.12)	–	–
Button	0.06 (0.05)	-0.01 (0.01)	-0.06 (0.03)*	0.00 (0.05)	2.86 (3.06)	1.12 (1.82)
Dialog	0.18 (0.04)***	-0.03 (0.01)*	-0.07 (0.02)**	-0.08 (0.04)*	0.36 (1.58)*	0.32 (1.46)**
Fullscreen	0.33 (0.08)***	-0.08 (0.02)**	-0.20 (0.05)***	-0.05 (0.08)	0.03 (6.23)	–
OnProtect	0.20 (0.10)	-0.04 (0.03)	-0.15 (0.06)*	-0.01 (0.10)	0.15 (3.19)	–
Overlay	0.41 (0.08)***	-0.06 (0.02)*	-0.19 (0.05)***	-0.16 (0.08)	0.17 (2.92)	1.04 (3.29)
Text	0.19 (0.04)***	-0.01 (0.01)	-0.08 (0.02)**	-0.11 (0.04)**	0.21 (1.90)*	0.68 (1.57)
After						
Banner	0.15 (0.04)***	-0.03 (0.01)*	-0.02 (0.03)	-0.10 (0.04)*	0.53 (1.84)	1.06 (1.48)
Button	0.23 (0.11)*	-0.04 (0.03)	0.01 (0.07)	-0.21 (0.11)	0.05 (4.90)	0.58 (2.36)
Dialog	0.15 (0.04)***	-0.02 (0.01)*	-0.05 (0.02)*	-0.07 (0.04)	0.23 (1.67)**	0.45 (1.55)
Fullscreen	-0.06 (0.09)	0.00 (0.03)	0.08 (0.06)	-0.03 (0.09)	29.08 (7.32)	–
OnProtect	0.15 (0.07)*	-0.03 (0.02)	-0.01 (0.04)	-0.12 (0.07)	1.13 (2.16)	0.57 (1.77)
Overlay	0.08 (0.09)	-0.03 (0.03)	-0.04 (0.06)	-0.01 (0.09)	0.18 (5.16)	0.61 (4.90)
SideMap	0.15 (0.07)*	-0.03 (0.02)	-0.05 (0.04)	-0.07 (0.07)	0.20 (2.69)	0.63 (2.12)
Text	0.09 (0.04)*	-0.04 (0.01)**	-0.03 (0.03)	-0.03 (0.04)	0.68 (3.13)	0.33 (2.14)
Additional						
Prompt	-0.09 (0.07)	0.02 (0.02)	0.00 (0.04)	0.06 (0.07)	1.79 (3.00)	1.40 (2.46)
Alternat.	-0.09 (0.04)*	0.02 (0.01)	0.02 (0.02)	0.05 (0.04)	2.89 (2.01)	0.71 (1.67)

that provided reassurance about how collected data would be used or that explained a delay caused by the device also increased grant rates (+11%). However, our dataset contained only a limited number of such examples.

6.7.1.2 The Effect of Text Attributes on User Sentiment.

We also built logistic regression models for the user sentiment dataset (see Section 6.3.4). These models showed no significant effects of the rationale text attributes on whether permission prompts were perceived as annoying or easy to decide on.

6.7.2 Analysis of Rationale UI Patterns

For the following analysis, we categorized the identified rationale patterns into two groups: those displayed before or alongside a browser prompt and those shown after permission is not granted (i.e., dismissed, ignored, or denied). This is a central distinction

as some users may never see the rationales shown after the interaction with the prompt. It should also be noted that some rationales are consistently displayed throughout the permission request cycle, remaining static. This particularly applies to text embedded within the main content of the webpage (referred to as *Before: Text* in Table 6.10) or to buttons that trigger a permission-protected function (referred to as *Before: Button*).

We also introduced two additional variables: *Alternative*, indicating whether the rationale includes an alternative to granting permission, and *Prompt*, indicating whether the rationale is displayed at the same time as the browser prompt. We then conducted a logistic regression on the patterns, as detailed in Table 6.10, using the methodology outlined in Section 6.3.4.

6.7.2.1 The Effect of UI Patterns on Permission Prompt Actions.

Similarly to the rationale texts, our exploratory analysis of rationale UIs showed a consistent pattern: the presence of rationales generally increases grant rates while reducing deny, ignore, and dismiss rates. The primary distinction between different rationale patterns lies in the magnitude of their effects.

Insight #1: Timing. Rationales presented before or alongside a browser prompt had a stronger impact than those shown after permission was not granted. In this context, overlays resulted in the highest increase in grants (+41%), followed by fullscreens (+33%), text (+19%), and dialogs (+18%). The largest reductions in deny rates were observed with fullscreen prompts (-8%), followed by overlays (-6%), before or alongside a browser prompt. Similar effects were noted for dismiss rates, with fullscreen (-20%), overlays (-19%), and rationales on permission-protected content (-15%) significantly decreasing the likelihood of dismissing a permission request. Regarding ignore rates, significant reductions were seen with text (-11%) and dialogs (-8%) presented before or alongside prompts, and banners shown after permission was not granted (-10%).

Insight #2: Actionable Buttons. For websites that offeed rationales after permission was not granted, the button layout resulted in the highest increase in grant rates (+23%) by offering users an actionable option to grant permission after experiencing the site without the requested permission.

Insight #3: Alternative. When a rationale offered users an alternative option to granting, the likelihood of users granting the requested permission decreased by 9%.

6.7.2.2 The Effect of UI Patterns on User Sentiment.

In analyzing the user experience data, we found that only dialogs and text significantly impacted user perception. Users were more likely to report an increase in annoyance when a rationale was presented as a dialog before a browser prompt and after. Text before a prompt was also associated with increased annoyance. Additionally, dialogs before a prompt made it less likely that respondents rated the decision-making process as somewhat or very easy.

6.8 Summary and Discussion

We discuss threats to the validity, summarize our main findings, and outline their broader implications.

6.8.1 Threats to Validity

We relied on web crawling to collect snapshots of webpages and their associated permission rationales. However, crawling is a challenging task [146, 38] and we may have missed pages containing permission rationales, such as rationales behind user authentication and those presented exclusively to specific geographical regions or specific web clients like mobile browsers. Furthermore, we focused on rationales based on English text. Consequently, our findings likely represent a lower-bound estimate of rationale prevalence on the web and may have missed mobile-specific, geo-specific, and patterns that depend on more complex user interactions.

In addition, to assess the effects of rationales on user decisions, we relied on Chrome telemetry and user sentiment data. However, telemetry data may not always correspond to the specific rationales we identified in webpages. Also, these were collected almost 1.5 years after our data collection, posing risks that some webpages may have changed in the meantime. Future research could build on our work and address these challenges by integrating more advanced crawling techniques and by conducting controlled and longitudinal studies to capture evolving web content over time.

Furthermore, future work could investigate whether libraries diverge significantly from the observed patterns and effects, and assess how their rationale text and design influences permission decisions. Our dataset does not allow us to fully isolate library and custom rationales at scale, as webpages may use both or customize library rationales, which complicates analyzing their distinct effects.

Then, we only assess how rationales observed on websites in the wild impact user behavior and sentiment. Therefore, we have a limited number of website samples and user sentiment responses for each of the various dimensions we identified. The websites in our sample also span a wide variety of use cases, given the differing nature of the most common permission-gated web APIs. It is therefore likely that properties of these use cases as well as other aspects such as brand reputation influenced the efficacy of the rationales we found. A controlled experiment across the dimensions we identified is necessary to more rigorously establish which types of rationales are truly effective. Such an experiment should also include a more thorough evaluation of user sentiment towards such rationales, given that our sentiment dataset was limited to only two of many plausible measures.

Finally, a validity threat may stem from users who never saw Chrome's permission prompt due to lack of interaction with the rationale presented on the webpage, leaving them unaccounted for in Chrome telemetry and experience sampling responses. This exclusion may bias the reported action rates and sentiment proportions, further highlighting the need for controlled experiments.

6.8.2 Open Science

To support and encourage future research, we have made our catalogs of rationale text and UI publicly accessible [42].

6.8.3 Web Permission Rationales

Detection Technique and Rationale Catalog. We present the first approach to systematically detect and study web permission rationales at scale, leveraging interactive web crawling, advanced semantic capabilities of LLMs, and BERT classification models. We instantiated our system against 779K webpages and created a comprehensive catalog of 3.6K unique, manually-vetted rationale text samples and 749 UIs. Furthermore, we found that the most common rationale messages are associated with 10 specific libraries, for which we developed 32 code signatures and HTML detection rules.

Status-Quo and Prevalence. Our automated crawler observed over 1.6M permission API calls on the surface of websites, accounting for over 162K webpages and 29.1K sites, with the majority belonging to geolocation and notification permission prompts. In total, our ML-based rationale detection pipeline, combined with mining of library signatures, identified over 85K webpages that present either a custom or library rationale.

6.8.4 The Effect of Rationales on User Decision-Making

We present and discuss the key insights from our study on how rationale text and UI elements influence user decisions regarding web permission prompts.

Insight #1: Timing is Everything–Early Rationales Drive Grants. Our results show that the timing of rationales is critical in influencing user decisions, which is in line with prior findings for rationales in mobile apps [P1]. Rationales shown before or alongside browser prompts increase the likelihood of users granting permissions. Overlays displayed at this stage resulted in the highest boost in grant rates (+41%), followed by fullscreens (+33%). These early interventions effectively set the stage for a positive user response, underscoring the importance of timing in permission request strategies. At the same time, those most effective rationales take up a large part of the screen, so can feel very heavy handed and might not be suitable for all types of permission use cases, especially when a capability is not a central part of the user journey.

Insight #2: Second Chances–Post-Prompt Buttons Can Be Helpful. Interestingly, we found that offering users actionable options after not granting a permission can substantially increase the likelihood of grants overall. Buttons presented in such situations were associated with 23% higher grant rates, giving users a second chance to reconsider their decision after experiencing the site without the requested permission. This finding highlights the value of providing users with a clear path to revisiting their initial permission choices.

Insight #3: Consequence-Based Messaging Less Effective. Our findings indicate that consequence-based rationales–those that emphasize what users stand to lose if they do not grant permission–are less effective and can even backfire. These strategies were associated with a 20% decrease in grant rates and a 12% increase in dismissals in our dataset. This suggests that emphasizing negative outcomes may undermine user trust and lead to resistance rather than compliance. Encouragement strategies should, therefore, focus on positive reinforcement rather than fear-based tactics.

Insight #4: Balancing User Annoyance and Effectiveness. While many rationales effectively increased grant rates, we also observe potentially unintended consequences on user experience. For example, text and dialog rationales presented before or alongside prompts were generally effective in increasing grant rates (+19% and +18%, respectively) and reducing dismiss rates. However, the dialogs included in our sample were also associated with higher levels of user annoyance. Similarly, text presented before a prompt made it more likely that user rated the decision-making process as more challenging. These findings suggest that while rationales are crucial for guiding user behavior, the challenge lies in balancing effectiveness with user satisfaction, ensuring that rationales are both persuasive and user-friendly.

Insight #5: Clarity on Essential Functions Drives Compliance. Our study also suggests that clearly stating the necessity of a functionality in the rationale text significantly improves grant rates (+8%) and reduces the likelihood of users ignoring the prompt (-5%). Users respond positively when they understand that granting a permission is essential for the core functionality of the site, as already posited by prior work [71]. This highlights the importance of clear, direct communication in rationale messages, particularly when the permission is crucial for the website's operation.

Insight #6: Message Tone Matters. Including a rationale message, even with a neutral or negative tone, improves grant rates and reduces dismiss and deny rates. Neutral tones increase grants by 8%, while positive tones increase them by 18%.

Insight #7: Effective Encouragement. Encouragement strategies play a critical role, with the use of consequences decreasing grant rates by 20% and raising dismiss rates by 12%. Conversely, highlighting benefits and motivations has a slight negative impact, lowering grant rates by 3% and increasing deny rates by 2%.

Insight #8: Functionality Requirements. Stating that functionality is required in the rationale results in an 8% higher grant rate and a 5% reduction in ignored prompts. Mentioning optional functionalities in text, however, does not significantly affect user actions. Yet, when there is an alternative interaction available, we found that grant rates are reduced by 9% in our UI-based analysis. This highlights that some users prefer alternative options when offered.

Insight #9: Message Content Impact. Providing additional context in rationale messages increases grant rates by 3% to 30%. Neutral permission requests have the strongest effect, while reassurances on data use and and notifications about device delays increase grant rates by 11%, though they were less common in our dataset.

6.8.5 Differences in Permission Requests Between Web and Android

In this study, we focused on permission experiences on the web when using desktop platforms, while prior work primarily addressed mobile app permissions. When comparing permission requests between the desktop web and mobile apps, web prompts offer users more interaction options. Users can grant or deny permissions, as well as ignore or dismiss requests. In contrast, Android or iOS permission prompts are inherently blocking, meaning the app's execution is paused until the user responds. Dismissing a prompt in the web context is loosely comparable to using the back button on Android while this behavior is not possible on iOS.

Both platforms exhibit similar rationale layouts, but Android's blocking prompts introduce distinct differences. On the desktop web, rationales can be displayed alongside permission requests due to the larger screen sizes, whereas on Android, they are shown either before or after a prompt. Additionally, limited screen space on Android made it less common to include supplementary content next to a rationale. For example, previous studies [P2] did not encounter rationales placed beside maps or inline text and banners, which typically appeared after permission was not granted.

Interestingly, when comparing our findings with previous work [P2], the phrasing and content of rationales across the two platforms were largely consistent. Most rationales aimed to encourage users to grant permissions, with 50.9% doing so on the desktop web compared to 67.0% on Android. A smaller proportion provided guidance (7.7% on the desktop web vs. 24.0% on Android). On both platforms, rationales emphasized the benefits of granting permissions to motivate users, with 10.6% of desktop web rationales and 18.0% of Android rationales highlighting this aspect. Less common themes, such as privacy assurances (1.6% vs. 2.0%) and alternative options (0.9% vs. 2.0%), followed similar trends. The main differences were in terminology, reflecting platform-specific contexts. For example, web rationales referred to *"websites"* and *"browser settings,"* while Android rationales mentioned *"apps"* and *"app settings."* A web rationale might state, *"To start your webcam, you need to allow our website to use it,"* whereas an Android rationale might say, *"We need access to your camera for the app to function."*

Finally, some capabilities, like geolocation, vary in usefulness depending on the attributes of devices typically used on the respective platforms. This influences how developers approach permission requests and design their features. For example, websites accessed mainly on desktop devices might provide alternative ways to locate the nearest store, as desktops often lack GPS sensors and provide lower-quality location data. Desktops also tend to stay in one place, making frequent location updates less relevant. In contrast, mobile devices like smartphones almost always have GPS sensors, offering high-quality location data and supporting features that rely on frequent updates of the current location, such as navigation.

6.8.6 Rationales and Dark Patterns

Dark patterns refer to design strategies in user interfaces that manipulate or deceive users into making decisions that may not align with their best interests [27]. These patterns exploit cognitive biases, making it harder for users to choose desirable options while subtly promoting unfavorable ones. We observed a tendency for such patterns in permission rationales, where the design may not fully qualify as a dark pattern but appears to nudge users toward a specific choice.

In our investigation of rationales, we noticed that websites often emphasize the "Allow" button, a practice also observed in Android rationales [P2]. Additionally, visual cues such as arrows pointing to the "Allow" button in browser prompts, as shown in Figure 6.11a, can serve as nudges. Attention icons may also draw focus to rationales, creating a sense of urgency. A particularly interesting case we found involved a rationale with a countdown timer, which caused the rationale to disappear after a set time. While it is not definitively a dark pattern, the countdown could influence users to make quick, potentially uninformed decisions. These observations highlight the importance of understanding these patterns to evaluate the ethical implications of such practices and to design user interfaces that support informed decision-making.

6.8.7 Decoupling Rationale Detection from Permission Prompts

Our crawler triggers about 20% of permission prompts in the Chrome telemetry dataset—a 100% improvement over non-interactive agents via our interaction heuristics (Section 10.1.1). This rate should not be mistaken for successfully identified rationales, the study's primary focus. Many applications present prompts without any rationale. Furthermore, since the ML pipeline relies on rationale text, it can detect rationales whenever the text is present in the DOM of webpages, regardless of whether the crawler triggers the prompt or not. In contrast, when rationales only appear in the DOM after triggering a prompt and the crawler cannot simulate the required user interactions, our approach will miss them. Quantitatively speaking, in 1000 random pages from telemetry, 113 had rationales following manual analysis, of which only 19 required user interaction for authentication. Accordingly, the crawler can detect rationales for over 80% of the pages presenting a rationale. Our results only provide a lower bound on the prevalence of rationales on the web. We refer interested readers to Appendix 10.1.2 for more details.

6.8.8 ML-based Rationale Detection and Future Work

Our study provides a systematic ML-based framework for detecting and analyzing permission rationales, offering a strong foundation for similar large-scale studies. For example, future work can use our tool and dataset of permission rationales to design and create more powerful classifiers. While our methodology is reusable, we acknowledge areas for development and encourage future research to build upon our findings.

Future research should explore the integration of complementary methods, such as signature-based detection of third-party libraries, to capture rationales not easily identified by text-based ML approaches. Expanding the method to automatically detect

non-textual rationale UIs (e.g., image processing) and incorporating multimodal learning could further enhance detection capabilities. To advance towards a fully automated process, researchers should also focus on integrating LLM-based crawlers able to complete tasks, such as simulating complex user actions [147], which could improve rationale coverage. While our pipeline effectively detects DOM-present rationales, manual analysis was necessary for validating the context of extracted sentences (e.g., distinguishing between tutorial text and true rationales). Future research should investigate automated methods, such as more powerful LLMs, to discern the context of potential rationales, enhancing the accuracy of the rationale detection pipeline.

6.8.9 Concluding Remarks

In this study, we adopted a predominantly quantitative approach to identify and analyze web permission rationales in the wild. Our findings indicate that web rationales do influence user behavior, though a complete list of possible effects is still unknown. To gain a more complete understanding, additional qualitative studies and controlled experiments are needed. For example, investigating the role of third-party libraries and their rationales is crucial for a thorough assessment of rationales' influence on user decisions. Similarly, more rigorously establishing the effects of rationale presentation based on the patterns we identified needs additional, controlled studies. In the meantime, our findings already provide actionable insights for researchers and practitioners into the diverse rationales present in the web ecosystem and their attributes.

7
Conclusion

Requesting users' permission to access sensitive data is a well-established practice in the mobile domain and is becoming increasingly relevant in the web domain. This approach enables more feature-rich and personalized experiences, for example, automatically determining a user's location to suggest nearby stores or enabling video calls. However, it also places a significant burden on users, who must make informed privacy-related decisions, often without sufficient context or understanding.

In this dissertation, we conducted a detailed analysis of the current runtime permission request models in both the mobile and web ecosystems. A central focus of this research was the role of permission rationales. We demonstrated that rationales can significantly influence not only the decision to grant or deny permission but also users' satisfaction with their choice, their perceived sense of control, and the extent to which their decision was informed.

Our findings highlight that users respond differently depending on how rationales are phrased. For example, a user may feel comfortable and satisfied in granting geolocation access because the rationale states that the permission is needed to find restaurants nearby and promises not to misuse the permission. The same user, however, might deny the same permission request and report feeling less informed and in control when the rationale states that "permission is needed for the app/website to function properly."

Beyond these clear contrasts, we also identified more subtle design factors that influence users' responses. For example, the articulation of the rationale (positive vs. negative phrasing), the visual layout of the message, and the timing of when the rationale is presented (before, during, or after the permission prompt) all play a role in shaping user behavior. These insights support the argument that rationales are a promising path forward for improving the usability and transparency of runtime permission requests. However, this research also raises two key challenges for future work:

Challenge #1: How can we ensure the integrity of rationales? In our study, we assumed that developers act in good faith and use rationales to support informed decision-making. However, in practice, some developers may be incentivized to craft misleading or coercive rationales to increase permission grant rates. Addressing this issue requires not only technical solutions but also a deeper understanding of user perception. One promising direction is the development of automated rationale vetting systems, which could use static or dynamic code analysis to assess whether a rationale's stated purpose aligns with actual app behavior. Similarly, requiring apps to declare explicit permission-functionality mappings during the review process could enable independent verification by app stores or browser vendors. Users could also contribute by flagging rationales they find suspicious.

Our research contributes concrete foundations for these solutions by uncovering how users interpret rationales and what factors influence their trust and decision-making. For instance, our findings highlight specific language patterns and contextual factors such as the timing of the rationale and the surrounding UI or app behavior that users associate with trustworthy explanations. These insights can directly support the design of automated vetting tools by identifying linguistic warning signs, detecting poorly timed or contextually inconsistent explanations, and informing platform policies on

acceptable rationale content. Moreover, by clarifying how users respond to different types of rationales, our work offers empirical grounding for transparency and fairness in permission requests.

Challenge #2: Should permission rationales be standardized? Given the nuanced impact of rationales, there is a compelling case for standardization. A promising direction would be to develop a modular rationale framework where developers choose from a curated set of pre-approved rationale components (e.g., purpose, benefit, data usage, and layout). This framework would help ensure clarity, reduce ambiguity, and support better decision-making across apps and websites. However, care must be taken to balance standardization with the flexibility developers need to accurately describe their app's functionality.

Our research offers valuable guidance for designing such a framework. We provide empirical findings on the building blocks commonly used in rationales and how users interpret different combinations of these elements. We identified specific language patterns and structural features that support user trust and understanding. These insights can inform which components should be included in a standardized system, how they should be phrased, and when they should be presented. When incorporating these evidence-based design principles into a modular rationale framework, platforms can improve the effectiveness of rationales and promote transparency in a scalable and user-centered manner.

In conclusion, this dissertation demonstrates that well-designed permission rationales can significantly enhance users' ability to make informed and confident privacy decisions. By examining how content, timing, phrasing, and presentation shape user perceptions and actions, this research contributes to a more user-centered understanding of the runtime permission model. The findings point to both immediate opportunities for improving current systems and longer-term challenges that require further exploration, particularly around ensuring honesty and consistency in rationales. As digital domains continue to rely on personal data, empowering users with transparent and meaningful permission requests will be a critical step toward building trust and accountability in both mobile and web platforms.

Bibliography

Author's Papers for this Thesis

[P1] **Elbitar, Y.**, Schilling, M., Nguyen, T. T., Backes, M., and Bugiel, S. Explanation beats context: the effect of timing & rationales on users' runtime permission decisions. In: *Proc. 30th USENIX Security Symposium (SEC'21)*. 2021.

[P2] **Elbitar, Y.**, Hart, A., and Bugiel, S. The power of words: a comprehensive analysis of rationales and their effects on users' permission decisions. In: *32nd Annual Network and Distributed System Security Symposium, (NDSS'25)*. 2025.

[P3] **Elbitar, Y.**, Khodayari, S., Harbach, M., De Stefano, G., Engedy, B. C., Pellegrino, G., and Bugiel, S. Permission rationales in the web ecosystem: an exploration of rationale text and design patterns. In: *Conference on Human Factors in Computing Systems (CHI'25)*. 2025.

Other references

[1] Acheampong, F. A., Nunoo-Mensah, H., and Chen, W. Transformer models for text-based emotion detection: a review of bert-based approaches. *Artificial Intelligence Review* 54 (2021), 5789–5829.

[2] Aguinis, H. and Bradley, K. J. Best practice recommendations for designing and implementing experimental vignette methodology studies. *Organizational Research Methods* 17 (2014), 351–371.

[3] Aguinis, H., Gottfredson, R. K., and Joo, H. Best-practice recommendations for defining, identifying, and handling outliers. *Organizational Research Methods* 16 (2013), 270–301.

[4] AI, M. *Mistral 7B Model*. Accessed: 2025-08-01.

[5] Akgul, O., Abu-Salma, R., Bai, W., Redmiles, E. M., Mazurek, M. L., and Ur, B. From secure to military-grade: exploring the effect of app descriptions on user perceptions of secure messaging. In: *Proc. 20th Workshop on Privacy in the Electronic Society (WPES'21)*. 2021.

[6] Akhawe, D. and Porter Felt, A. Alice in warningland: A large-scale field study of browser security warning effectiveness. In: *22th USENIX Security Symposium (SEC'13)*. 2013.

[7] Allen, M. J. and Yen, W. M. *Introduction to Measurement Theory*. Waveland Press, 2002.

[8] Almuhimedi, H., Schaub, F., Sadeh, N. M., Adjerid, I., Acquisti, A., Gluck, J., Cranor, L. F., and Agarwal, Y. Your location has been shared 5, 398 times!: A field study on mobile app privacy nudging. In: *Conference on Human Factors in Computing Systems (CHI'15)*. 2015.

[9] Apple. *About privacy and Location Services on iOS, iPadOS and watchOS*. https://support.apple.com/en-gb/102515. Accessed: 2025-08-01.

[10] Apple. *App Store review policy – privacy*. Accessed: 2025-08-01.

[11] Apple. *Privacy*. https://developer.apple.com/design/human-interface-guidelines/privacy. Accessed: 2025-08-01.

[12] Apple. *Privacy*. https://developer.apple.com/design/human-interface-guidelines/privacy#Pre-alert-screens-windows-or-views. Accessed: 2025-08-01.

[13] Apple. *Transparency is the best policy*. https://www.apple.com/privacy/labels. Accessed: 2025-08-01.

[14] Apple Developer Guide. *Information Property List*. https://developer.apple.com/documentation/BundleResources/Information-Property-List. Accessed: 2021-05-26.

[15] Apple Developer Guide. *Requesting access to protected resources*. https://developer.apple.com/documentation/uikit/protecting_the_user_s_privacy/requesting_access_to_protected_resources. Accessed: 2021-05-26.

[16] Apple Developer Guide. *Requesting permissions*. https://developer.apple.com/design/human-interface-guidelines/ios/app-architecture/requesting-permission/. Accessed: 2021-05-26.

[17] Arntz, P. *Browser push notifications: a feature asking to be abused*. https://blog.malwarebytes.com/security-world/technology/2019/01/browser-push-notifications-feature-asking-abused. Accessed: 2025-08-01.

[18] Bahattacherjee, A. *Social science research: Principles, methods and practices (2nd ed.)* Global text project, 2012.

[19] Bates, D., Mächler, M., Bolker, B., and Walker, S. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software* 67 (2015), 1–48.

[20] Bilogrevic, I., Engedy, B., Porter III, J. L., Taft, N., Hasanbega, K., Paseltiner, A., Lee, H. K., Jung, E., Watkins, M., McLachlan, P., et al. "shhh... be quiet!" reducing the unwanted interruptions of notification permission prompts on chrome. In: *Proc. 30th USENIX Security Symposium (SEC'21)*. 2021.

[21] Birnbaum, M. How to show that $9 > 221$: Collect judgments in a between-subjects design. *Psychological Methods* 4 (1999), 243–249.

[22] Bogdanas, D. Dperm: Assisting the migration of Android apps to runtime permissions. *CoRR* (2017).

[23] Bongard-Blanchy, K., Sterckx, J., Rossi, A., Distler, V., Rivas, S., and Koenig, V. An (un)necessary evil - users' (un)certainty about smartphone app permissions and implications for privacy engineering. In: *Proc. 7th IEEE European Symposium on Security and Privacy (EuroS&P'22)*. 2022.

[24] Bonné, B., Peddinti, S. T., Bilogrevic, I., and Taft, N. Exploring decision making with android's runtime permission dialogs using in-context surveys. In: *13th Symposium on Usable Privacy and Security (SOUPS'17)*. 2017.

[25] Brandimarte, L., Acquisti, A., and Loewenstein, G. Misplaced confidences: privacy and the control paradox. *Social Psychological and Personality Science* 4 (2013), 340–347.

[26] Bravo-Lillo, C., Komanduri, S., Cranor, L. F., Reeder, R. W., Sleeper, M., Downs, J. S., and Schechter, S. E. Your attention please: Designing security-decision UIs to make genuine risks harder to ignore. In: *9th Symposium on Usable Privacy and Security (SOUPS'13)*. 2013.

[27] Brignull, H. *Deceptive Patterns*. Accessed: 2025-08-01.

[28] Cao, W., Xia, C., Peddinti, S. T., Lie, D., Taft, N., and Austin, L. M. A large scale study of user behavior, expectations and engagement with android permissions. In: *Proc. 30th USENIX Security Symposium (SEC'21)*. 2021.

[29] Chen, S., Fan, L., Chen, C., and Liu, Y. Automatically distilling storyboard with rich features for android apps. *CoRR* abs/2203.06420 (2022).

[30] Chen, S., Fan, L., Chen, C., Su, T., Li, W., Liu, Y., and Xu, L. Storydroid: automated generation of storyboard for android apps. In: *Proc. 41th International Conference on Software Engineering (ICSE'19)*. 2019.

[31] Clelland, I. *W3C Working Draft: Permissions Policy*. `https://www.w3.org/TR/permissions-policy`. Accessed: 2025-08-01.

[32] Cowan, N. The magical number 4 in short-term memory: a reconsideration of mental storage capacity. *Behavioral and Brain Sciences* 24 (2001), 87–114.

[33] Cranor, L. F. Mobile-app privacy nutrition labels missing key ingredients for success. *Commun. ACM* 65 (2022), 26–28.

[34] Cumming, G. and Finch, S. Inference by eye: Confidence intervals and how to read pictures of data. *American psychologist* 60 (2005), 170.

[35] Cunningham, E. *Improving app security and performance on Google Play*. `https://android-developers.googleblog.com/2017/12/improving-app-security-and-performance.html`. Accessed: 2021-05-26.

[36] Dinev, T., Xu, H., Smith, H. J., and Hart, P. J. Information privacy and correlates: An empirical attempt to bridge and distinguish privacy-related concepts. *European Journal of Information Systems* 22 (2013), 295–316.

[37] Distler, V., Gutfleisch, T., Lallemand, C., Lenzini, G., and Koenig, V. Complex, but in a good way? how to represent encryption to non-experts through text and visuals – evidence from expert co-creation and a vignette experiment. *Computers in Human Behavior Reports* 5 (2022), 100161.

[38] Doupé, A., Cavedon, L., Kruegel, C., and Vigna, G. Enemy of the state: A state-aware black-box web vulnerability scanner. In: *Proc. 21st USENIX Security Symposium (SEC'12)*. 2012.

[39] Egelman, S., Cranor, L. F., and Hong, J. I. You've been warned: an empirical study of the effectiveness of web browser phishing warnings. In: *Conf. on Human Factors in Computing Systems (CHI'08)*. 2008.

[40] Elbitar, Y., Hart, A., and Bugiel, S. *Investigating the Effect of Rationales*. `https://osf.io/7zprb`. Accessed: 2025-08-01.

[41] Elbitar, Y., Hart, A., and Bugiel, S. *Rationale Codebook Landscape*. `https://osf.io/z9huf`. Accessed: 2025-08-01.

[42] Elbitar, Y., Khodayari, S., Harbach, M., Stefano, G. D., Engedy, B. C., Pellegrino, G., and Bugiel, S. *Catalogs of rationale text and UI*. Accessed: 2025-08-01.

[43] Explosion. *Prodigy*. `https://prodi.gy`. Accessed: 2025-08-01.

[44] Explosion. *SpaCy*. `https://spacy.io`. Accessed: 2025-08-01.

[45] Face, H. *BERT Base Model*. Accessed: 2025-08-01.

[46] Face, H. *Sentence Transformer all-MiniLM-L6-v2*. Accessed: 2025-08-01.

[47] Faul, F., Erdfelder, E., Buchner, A., and Lang, A.-G. Statistical power analyses using g*power 3.1: Tests for correlation and regression analyses. *Behavior research methods* 41 (2009), 1149–60.

[48] Felt, A. P., Ha, E., Egelman, S., Haney, A., Chin, E., and Wagner, D. A. Android permissions: user attention, comprehension, and behavior. In: *8th Symposium on Usable Privacy and Security (SOUPS'12)*. 2012.

[49] Feng, Y., Chen, L., Zheng, A., Gao, C., and Zheng, Z. Ac-net: assessing the consistency of description and permission in android apps. *IEEE Access* 7 (2019), 57829–57842.

[50] Frey, S. *New safety section in Play will give transparency into how apps use data*. `https://android-developers.googleblog.com/2021/05/new-safety-section-in-google-play-will.html`. Accessed: 2021-05-26.

[51] Funder, D. C. and Ozer, D. J. Evaluating effect size in psychological research: sense and nonsense. *Advances in Methods and Practices in Psychological Science* 2 (2019), 156–168.

[52] Gao, H., Guo, C., Wu, Y., Dong, N., Hou, X., Xu, S., and Xu, J. Autoper: Automatic recommender for runtime-permission in Android applications. In: *Proc. 43rd IEEE Annual Computer Software and Applications Conference (COMPSAC'19)*. 2019.

[53] Gardner, J., Feng, Y., Reiman, K., Lin, Z., Jain, A., and Sadeh, N. Helping mobile application developers create accurate privacy labels. In: *Proc. 7th IEEE European Symposium on Security and Privacy (EuroS&P'22)*. 2022.

[54] Garrido-Merchan, E. C., Gozalo-Brizuela, R., and Gonzalez-Carvajal, S. Comparing bert against traditional machine learning models in text classification. *Journal of Computational and Cognitive Engineering* 2 (2023), 352–356.

[55] Gasparis, I., Aqil, A., Qian, Z., Song, C., Krishnamurthy, S. V., Gupta, R., and Colbert, E. Droid M+: Developer support for imbibing Android's new permission model. In: *Asia Conference on Computer and Communications Security (AsiaCCS'18)*. 2018.

[56] Geldhof, G. J., Preacher, K. J., and Zyphur, M. J. Reliability estimation in a multilevel confirmatory factor analysis framework. *Psychological methods* 19 (2014), 72–91.

[57] GlobalStats. *Mobile Operating System Market Share Worldwide*. `https://gs.statcounter.com/os-market-share/mobile/worldwide`. Accessed: 2025-08-01.

[58] Golla, M., Ho, G., Lohmus, M., Pulluri, M., and Redmiles, E. M. Driving 2fa adoption at scale: optimizing two-factor authentication notification design patterns. In: *Proc. 30th USENIX Security Symposium, (SEC'21)*. 2021.

[59] Google. *ActivityCompat*. `https://developer.android.com/reference/androidx/core/app/ActivityCompat#shouldShowRequestPermissionRationale(android.app.Activity,java.lang.String)`. Accessed: 2025-08-01.

[60] Google. *checkSelfPermission*. `https://developer.android.com/reference/androidx/core/content/ContextCompat#checkSelfPermission(android.content.Context,java.lang.String)`. Accessed: 2025-08-01.

[61] Google. *Dialogs*. `https://developer.android.com/develop/ui/views/components/dialogs`. Accessed: 2025-08-01.

[62] Google. *Google Play console help - Declaring permissions for your app*. Accessed: 2025-08-01.

[63] Google. *Localize your app*. `https://developer.android.com/guide/topics/resources/localization`. Accessed: 2025-08-01.

[64] Google. *One-time permissions*. `https://developer.android.com/training/permissions/requesting#one-time`. Accessed: 2025-08-01.

[65] Google. *onRequestPermissionsResult*. `https://developer.android.com/reference/androidx/core/app/ActivityCompat.OnRequestPermissionsResultCallback#onRequestPermissionsResult(int,java.lang.String[],int[])`. Accessed: 2025-08-01.

[66] Google. *Request runtime permissions*. `https://developer.android.com/training/permissions/requesting`. Accessed: 2025-08-01.

[67] Google. *requestPermissions*. `https://developer.android.com/reference/androidx/core/app/ActivityCompat#requestPermissions(android.app.Activity,java.lang.String[],int)`. Accessed: 2025-08-01.

[68] Google. *Understand app privacy & security practices with Google Play's Data safety section.* `https://support.google.com/googleplay/answer/11416267`. Accessed: 2025-08-01.

[69] Google. *Write automated tests with UI Automator.* `https://developer.android.com/training/testing/other-components/ui-automator`. Accessed: 2025-08-01.

[70] *Guidelines for academic requesters.* `https://www.yumpu.com/en/document/read/31225336/guidelines-for-academic-requesters`. Accessed: 2021-05-26.

[71] Harbach, M. Websites need your permission too – user sentiment and decision-making on web permission prompts in desktop chrome. In: *Conference on Human Factors in Computing Systems (CHI'24)*. 2024.

[72] Harbach, M., Bilogrevic, I., Bacis, E., Chen, S., Uppal, R., Paicu, A., Klim, E., Watkins, M., and Engedy, B. Don't interrupt me - A large-scale study of on-device permission prompt quieting in chrome. In: *31st Annual Network and Distributed System Security Symposium (NDSS'24)*. 2024.

[73] Harbach, M., Hettig, M., Weber, S., and Smith, M. Using personal examples to improve risk communication for security & privacy decisions. In: *Conf. on Human Factors in Computing Systems (CHI'14)*. 2014.

[74] Harbach, M. and Steiner, T. *Web permissions best practices.* Accessed: 2025-08-01.

[75] Harrison, R. Introduction to monte carlo simulation. *AIP conference proceedings* 1204 (2010), 17–21.

[76] Hazhirpasand, M., Ghafari, M., and Nierstrasz, O. Tricking johnny into granting web permissions. In: *Proc. 24th Evaluation and Assessment in Software Engineering (EASE'20)*. 2020.

[77] Hive, W. *Storerocket Library.* Accessed: 2025-08-01.

[78] Hoffman, A. *Smart Push Library.* Accessed: 2025-08-01.

[79] Hox, J., Moerbeek, M., and Schoot, R. van de. *Multilevel Analysis: Techniques and Applications (3nd ed.)* Routledge, 2017.

[80] Hughes, R. and Huby, M. The application of vignettes in social and nursing research. *Journal of advanced nursing* 37 (2002), 382–6.

[81] iBotPeaches. *Apktool.* `https://apktool.org`. Accessed: 2025-08-01.

[82] iZooto. *iZooto Library.* Accessed: 2025-08-01.

[83] Jain, A., Rodriguez, D., Álamo, J. M. del, and Sadeh, N. M. ATLAS: automatically detecting discrepancies between privacy policies and privacy labels. In: *Proc. 8th IEEE European Symposium on Security and Privacy (EuroS&P'23)*. 2023.

[84] Keith, M., Tay, L., and Harms, P. Systems perspective of Amazon Mechanical Turk for organizational research: Review and recommendations. *Frontiers in Psychology* 8 (2017), 1359.

[85] Kelley, P. G., Bresee, J., Cranor, L. F., and Reeder, R. W. A "nutrition label" for privacy. In: *Proc. 5th Symposium on Usable Privacy and Security (SOUPS'12)*. 2009.

[86] Kelley, P. G., Consolvo, S., Cranor, L. F., Jung, J., Sadeh, N. M., and Wetherall, D. A conundrum of permissions: installing applications on an android smartphone. In: *Proc. 16th International Conference on Financial Cryptography and Data Security (FC'12)*. 2012.

[87] Kelley, P. G., Cranor, L. F., and Sadeh, N. M. Privacy as part of the app decision-making process. In: *ACM SIGCHI Conference on Human Factors in Computing Systems (SIGCHI'13)*. 2013.

[88] Koch, S., Wessels, M., Altpeter, B., Olvermann, M., and Johns, M. Keeping privacy labels honest. *Proc. Priv. Enhancing Technol.* 2022 (2022), 486–506.

[89] Kollnig, K., Shuba, A., Kleek, M. V., Binns, R., and Shadbolt, N. Goodbye tracking? impact of ios app tracking transparency and privacy labels. In: *Proc. 5th ACM Conference on Fairness, Accountability, and Transparency (FAccT'22)*. 2022.

[90] Levin, I. P., Schneider, S. L., and Gaeth, G. J. All frames are not created equal: a typology and critical analysis of framing effects. *Organizational Behavior and Human Decision Processes* 76 (1998), 149–188.

[91] Li, T., Cranor, L. F., Agarwal, Y., and Hong, J. I. Matcha: an IDE plugin for creating accurate privacy nutrition labels. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8 (2024), 33:1–33:38.

[92] Li, T., Reiman, K., Agarwal, Y., Cranor, L. F., and Hong, J. I. Understanding challenges for developers to create accurate privacy nutrition labels. In: *Conference on Human Factors in Computing Systems (CHI'22)*. 2022.

[93] Li, Y., Yang, Z., Guo, Y., and Chen, X. Droidbot: A lightweight ui-guided test input generator for Android. In: *Proc. 39th International Conference on Software Engineering (ICSE'17)*. 2017.

[94] Li, Y., Chen, D., Li, T., Agarwal, Y., Cranor, L. F., and Hong, J. I. Understanding ios privacy nutrition labels: an exploratory large-scale analysis of app store data. In: *Conference on Human Factors in Computing Systems (CHI'22)*. 2022.

[95] Liccardi, I., Pato, J. N., Weitzner, D. J., Abelson, H., and Roure, D. D. No technical understanding required: Helping users make informed choices about access to their personal data. In: *11th International Conference on Mobile and Ubiquitous Systems (MOBIQUITOUS'14)*. 2014.

[96] Lin, J., Liu, B., Sadeh, N. M., and Hong, J. I. Modeling users' mobile app privacy preferences: Restoring usability in a sea of permission settings. In: *Proc. 10th Symposium on Usable Privacy and Security (SOUPS'14)*. 2014.

[97] Lin, J., Sadeh, N. M., Amini, S., Lindqvist, J., Hong, J. I., and Zhang, J. Expectation and purpose: Understanding users' mental models of mobile app privacy through crowdsourcing. In: *ACM Conference on Ubiquitous Computing, (Ubicomp'12)*. 2012.

[98] Lin, Y., Juneja, J., Birrell, E., and Cranor, L. F. Data safety vs. app privacy: comparing the usability of android and ios privacy labels. *CoRR* abs/2312.03918 (2023).

[99] Litman, L., Robinson, J., and Abberbock, T. Turkprime.com: A versatile crowdsourcing data acquisition platform for the behavioral sciences. *Behavior Research Methods* 49 (2017), 433–442.

[100] Liu, B., Andersen, M. S., Schaub, F., Almuhimedi, H., Zhang, S., Sadeh, N. M., Agarwal, Y., and Acquisti, A. Follow my recommendations: A personalized privacy assistant for mobile app permissions. In: *Proc. 12th Symposium on Usable Privacy and Security (SOUPS'16)*. 2016.

[101] Liu, B., Lin, J., and Sadeh, N. M. Reconciling mobile app privacy and usability on smartphones: could user privacy profiles help? In: *Proc. 23rd International World Wide Web Conference (WWW'14)*. 2014.

[102] Liu, X., Leng, Y., Yang, W., Wang, W., Zhai, C., and Xie, T. A large-scale empirical study on Android runtime-permission rationale messages. In: *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'18)*. 2018.

[103] Liu, X., Leng, Y., Yang, W., Zhai, C., and Xie, T. Mining android app descriptions for permission requirements recommendation. In: *26th IEEE International Requirements Engineering Conference (RE'18)*. 2018.

[104] Lorah, J. Effect size measures for multilevel models: definition, interpretation, and timss example. *Large-scale Assessments in Education* 6 (2018).

[105] Malhotra, N. K., Kim, S. S., and Agarwal, J. Internet users' information privacy concerns (IUIPC): The construct, the scale, and a causal model. *Information Systems Research* 15 (2004), 336–355.

[106] Material Design. *Android permissions*. `https://material.io/design/platform-guidance/android-permissions.html`. Accessed: 2021-05-26.

[107] Material Design. *Dialogs*. `https://material.io/components/dialogs`. Accessed: 2021-05-26.

[108] Material Design. *Writing*. `https://material.io/design/communication/writing.html`. Accessed: 2021-05-26.

[109] Micinski, K. K., Votipka, D., Stevens, R., Kofinas, N., Mazurek, M. L., and Foster, J. S. User interactions and permission use on android. In: *Conference on Human Factors in Computing Systems (CHI'17)*. 2017.

[110] Microsoft. *Reducing distractions with quiet notification requests*. `https://blogs.windows.com/msedgedev/2020/07/23/reducing-distractions-quiet-notification-requests`. Accessed: 2025-08-01.

[111] Mihalcea, R., Corley, C., and Strapparava, C. Corpus-based and knowledge-based measures of text semantic similarity. In: *Proc. 21st National Conference on Artificial Intelligence (AAAI'06)*. 2006.

[112] MoEngage. *Moe-push Library*. Accessed: 2025-08-01.

[113] Mozilla. *Geolocation API*. Accessed: 2025-08-01.

[114] Mozilla. *Restricting notification permission prompts in Firefox*. `https://blog.mozilla.org/futurereleases/2019/11/04/restricting-notification-permission-prompts-in-firefox`. Accessed: 2025-08-01.

[115] Mukherjee, D., Ahmadi, A., Pour, M. V., and Reardon, J. An empirical study on user reviews targeting mobile apps' security & privacy. *arXiv:2010.06371* (2020). Retrieved from https://arxiv.org/abs/2010.06371.

[116] Nguyen, D., Derr, E., Backes, M., and Bugiel, S. Short text, large effect: Measuring the impact of user reviews on android app security & privacy. In: *Proc. 30th IEEE Symposium on Security and Privacy (SP'19)*. 2019.

[117] Nielsen Norman Group. *3 Design Considerations for Effective Mobile-App Permission Requests*. `https://www.nngroup.com/articles/permission-requests`. Accessed: 2025-08-01.

[118] NLTK. *Categorizing and tagging words*. `https://www.nltk.org/book/ch05.html`. Accessed: 2021-05-26.

[119] Olejnik, K., Dacosta, I., Machado, J. S., Huguenin, K., Khan, M. E., and Hubaux, J. Smarper: Context-Aware and automatic runtime-permissions for mobile devices. In: *Proc. 28th IEEE Symposium on Security and Privacy (SP'17)*. 2017.

[120] OneSignal. *OneSignal Library*. Accessed: 2025-08-01.

[121] Pan, E., Ren, J., Lindorfer, M., Wilson, C., and Choffnes, D. R. Panoptispy: Characterizing audio and video exfiltration from android applications. *Proc. Priv. Enhancing Technol.* 2018 (2018), 33–50.

[122] Pan, X., Cao, Y., Du, X., He, B., Fang, G., Shao, R., and Chen, Y. Flowcog: Context-aware semantics extraction and analysis of information flow leaks in Android apps. In: *Proc. 27th USENIX Security Symposium, (SEC'18)*. 2018.

[123] Pandita, R., Xiao, X., Yang, W., Enck, W., and Xie, T. WHYPER: towards automating risk assessment of mobile applications. In: *Proc. 22th USENIX Security Symposium (SEC'13)*. 2013.

[124] Peddinti, S. T., Bilogrevic, I., Taft, N., Pelikan, M., Erlingsson, Ú., Anthonysamy, P., and Hogben, G. Reducing permission requests in mobile apps. In: *Proc. Internet Measurement Conference (IMC'19)*. 2019.

[125] Perfecty. *Perfecty Library*. Accessed: 2025-08-01.

[126] Porter Felt, A., Egelman, S., Finifter, M., Akhawe, D., and Wagner, D. A. How to ask for permission. In: *Proc. 7th USENIX Workshop on Hot Topics in Security (HotSec'12)*. 2012.

[127] Porter Felt, A., Egelman, S., and Wagner, D. A. I've got 99 problems, but vibration ain't one: a survey of smartphone users' concerns. In: *Proc. Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM'12)*. 2012.

[128] Prolific. *Prolific*. `https://www.prolific.com`. Accessed: 2025-08-01.

[129] PushEngage. *Pushengage Library*. Accessed: 2025-08-01.

[130] pushowl. *PushOWL Library*. Accessed: 2025-08-01.

[131] Qasim, R., Bangyal, W. H., Alqarni, M. A., and Ali Almazroi, A. A fine-tuned bert-based transfer learning approach for text classification. *Journal of healthcare engineering* 2022 (2022), 3498123.

[132] Qu, Z., Rastogi, V., Zhang, X., Chen, Y., Zhu, T., and Chen, Z. Autocog: Measuring the description-to-permission fidelity in Android applications. In: *Proc. 21st ACM Conference on Computer and Communications Security (SIGSAC'14)*. 2014.

[133] R Core Team. *R: A Language and Environment for Statistical Computing*. 2020.

[134] Richardson, L. *Beautiful Soup Library*. Accessed: 2025-08-01.

[135] Robinson, J., Rosenzweig, C., Moss, A. J., and Litman, L. Tapped out or barely tapped? Recommendations for how to harness the vast and largely unused potential of the Mechanical Turk participant pool. *PLOS ONE* 14 (2019), 1–29.

[136] Rodriguez, D., Jain, A., Alamo, J. M. D., and Sadeh, N. Comparing privacy label disclosures of apps published in both the app store and google play stores. In: *Proc. 8th IEEE European Symposium on Security and Privacy Workshops (EuroS&PW'23)*. 2023.

[137] Ruth, K., Kumar, D., Wang, B., Valenta, L., and Durumeric, Z. Toppling top lists: evaluating the accuracy of popular website lists. In: *Proc. ACM Internet Measurement Conference (IMC'22)*. 2022.

[138] Schechter, S. E., Dhamija, R., Ozment, A., and Fischer, I. The emperor's new security indicators. In: *18th IEEE Symposium on Security and Privacy (SP'07)*. 2007.

[139] scikit-learn. *Sklearn Agglomerative Clustering*. Accessed: 2025-08-01.

[140] Scoccia, G. L., Autili, M., Stilo, G., and Inverardi, P. An empirical study of privacy labels on the apple ios mobile app store. In: *Proc. 9th IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft '22)*. 2022.

[141] Sela, A., Wheeler, S. C., and Sarial-Abi, G. We are not the same as you and i: causal effects of minor language variations on consumers' attitudes toward brands. *Journal of Consumer Research* 39 (2012), 644–661.

[142] Shen, B., Wei, L., Xiang, C., Wu, Y., Shen, M., Zhou, Y., and Jin, X. Can systems explain permissions better? understanding users' misperceptions under smartphone runtime permission model. In: *Proc. 30th USENIX Security Symposium, (SEC'21)*. 2021.

[143] Shklovski, I., Mainwaring, S. D., Skúladóttir, H. H., and Borgthorsson, H. Leakiness and creepiness in app space: Perceptions of privacy and mobile app use. In: *Conference on Human Factors in Computing Systems (CHI'14)*. 2014.

[144] Smith, H. J., Milberg, S. J., and Burke, S. J. Information privacy: measuring individuals' concerns about organizational practices. *MIS Quarterly* 20 (1996), 167–196.

[145] Spector, P. E. *Research designs*. SAGE Publications, Inc., 1981.

[146] Stafeev, A. and Pellegrino, G. Sok: state of the krawlers-evaluating the effectiveness of crawling algorithms for web security measurements. In: *Proc. 33rd USENIX Security Symposium (SEC'24)*. 2024.

[147] Stafeev, A., Recktenwald, T., De Stefano, G., Khodayari, S., and Pellegrino, G. Yurascanner: leveraging llms for task-driven web app scanning. In: *32nd Annual Network and Distributed System Security Symposium, (NDSS'25)*. 2025.

[148] Stalmeier, P. F., Roosmalen, M. S., Verhoef, L. C., Hoekstra-Weebers, J. E., Oosterwijk, J. C., Moog, U., Hoogerbrugge, N., and van Daal, W. A. The decision evaluation scales. *Patient Education and Counseling* 57 (2005), 286–293.

[149] Steiger, A. and Kühberger, A. A meta-analytic re-appraisal of the framing effect. *Zeitschrift für Psychologie* 226 (2018), 45–55.

[150] Stevens, R., Ganz, J., Filkov, V., Devanbu, P. T., and Chen, H. Asking for (and about) permissions used by android apps. In: *Proc. 10th Working Conference on Mining Software Repositories (MSR'13)*. 2013.

[151] Stopper, A. and Caltrider, J. *See No Evil: Loopholes in Google's Data Safety Labels Keep Companies in the Clear and Consumers in the Dark*. `https://foundation.mozilla.org/en/campaigns/googles-data-safety-labels/`. Accessed: 2025-08-01.

[152] Subramani, K., Yuan, X., Setayeshfar, O., Vadrevu, P., Lee, K. H., and Perdisci, R. When push comes to ads: measuring the rise of (malicious) push advertising. In: *Proc. ACM Internet Measurement Conference (IMC'20)*. 2020.

[153] Sunshine, J., Egelman, S., Almuhimedi, H., Atri, N., and Cranor, L. F. Crying wolf: An empirical study of SSL warning effectiveness. In: *Proc. 18th USENIX Security Symposium, (SEC'09)*. 2009.

[154] SuperStoreFinder. *Superstorefinder-wp Library*. Accessed: 2025-08-01.

[155] Tahaei, M., Abu-Salma, R., and Rashid, A. Stuck in the permissions with you: developer & end-user perspectives on app permissions & their privacy ramifications. In: *Conference on Human Factors in Computing Systems (CHI'23)*. 2023.

[156] Tan, J., Nguyen, K., Theodorides, M., Negrón-Arroyo, H., Thompson, C., Egelman, S., and Wagner, D. A. The effect of developer-specified explanations for permission requests on smartphone user behavior. In: *Conference on Human Factors in Computing Systems (CHI'14)*. 2014.

[157] Thompson, C., Johnson, M., Egelman, S., Wagner, D., and King, J. When it's better to ask forgiveness than get permission: attribution mechanisms for smartphone resources. In: *Proc. 9th Symposium on Usable Privacy and Security (SOUPS'13)*. 2013.

[158] Tsai, L., Wijesekera, P., Reardon, J., Reyes, I., Egelman, S., Wagner, D. A., Good, N., and Chen, J. Turtle guard: Helping Android users apply contextual privacy preferences. In: *Proc. 13th Symposium on Usable Privacy and Security (SOUPS'17)*. 2017.

[159] Tversky, A. and Kahneman, D. The framing of decisions and the psychology of choice. *Science* 211 (1981), 453–458.

[160] Votipka, D., Rabin, S. M., Micinski, K., Gilray, T., Mazurek, M. L., and Foster, J. S. User comfort with android background resource accesses in different contexts. In: *Proc. 14th Symposium on Usable Privacy and Security (SOUPS'18)*. 2018.

[161] W3C. *Media Capture and Streams*. Accessed: 2025-08-01.

[162] W3C. *Web Push API*. Accessed: 2025-08-01.

[163] Wang, H., Hong, J. I., and Guo, Y. Using text mining to infer the purpose of permission use in mobile apps. In: *Proc. ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'15)*. 2015.

[164] Ward, M. and Meade, A. W. Dealing with careless responding in survey data: prevention, identification, and recommended best practices. *Annual Review of Psychology* 74 (2023), 577–596.

[165] Watson, K., Just, M., and Berg, T. A comic-based approach to permission request communication. *Comput. Secur.* 124 (2023), 102942.

[166] webpush3. *Webpushr Library*. Accessed: 2025-08-01.

[167] Wei, X., Gomez, L., Neamtiu, I., and Faloutsos, M. Permission evolution in the android ecosystem. In: *Proc. 28th Annual Computer Security Applications Conference, (ACSAC'12)*. 2012.

[168] WHATWG. *DOM Living Standard*. Accessed: 2025-08-01.

[169] WHATWG. *Web Notifications API*. Accessed: 2025-08-01.

[170] Wijesekera, P., Baokar, A., Hosseini, A., Egelman, S., Wagner, D. A., and Beznosov, K. Android permissions remystified: A field study on contextual integrity. In: *Proc. 24th USENIX Security Symposium (SEC'15)*. 2015.

[171] Wijesekera, P., Baokar, A., Tsai, L., Reardon, J., Egelman, S., Wagner, D. A., and Beznosov, K. The feasibility of dynamically granted permissions: Aligning mobile privacy with user preferences. In: *Proc. 28th IEEE Symposium on Security and Privacy (SP'17)*. 2017.

[172] Woehr, D. J. and Lance, C. E. Paper people versus direct observation: An empirical examination of laboratory methodologies. *Journal of Organizational Behavior* 12 (1991), 387–397.

[173] Xiao, Y., Li, Z., Qin, Y., Bai, X., Guan, J., Liao, X., and Xing, L. Lalaine: measuring and characterizing non-compliance of apple privacy labels. In: *Proc. 32nd USENIX Security Symposium (SEC'23)*. 2023.

[174] Zhang, B. and Xu, H. Privacy nudges for mobile applications: effects on the creepiness emotion and privacy attitudes. In: *Proc. 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW'16)*. 2016.

[175] Zhang, M., Duan, Y., Feng, Q., and Yin, H. Towards automatic generation of security-centric descriptions for Android apps. In: *Proc. 22nd ACM Conference on Computer and Communications Security (SIGSAC'15)*. 2015.

[176] Zhang, S., Feng, Y., Yao, Y., Cranor, L. F., and Sadeh, N. How usable are ios app privacy labels? *Proc. Priv. Enhancing Technol.* 4 (2022), 204–228.

[177] Zhang, S. and Sadeh, N. Do privacy labels answer users' privacy questions? *Workshop on Usable Security and Privacy* (2023).

8

Appendix: Timing & Rationales

8.1 Study Procedure

This section lists the questions of the survey in the same order they were shown to participants. Note that Sections 8.1.1 and 8.1.2 are repeated four times per participant.

8.1.1 Pre-Questionnaire

The *{first/second/third/last}* app of interest is called *{app name}*. Imagine the following scenario: You have recently installed the *{app name}* app on your phone. *{sentence describing the major functionalities of the app}*. You want to use this app to *{objective to use the app}*. [Show a screenshot of the homescreen with the app icon.]

App Familiarity: Have you used this app before?

- ○ Yes
- ○ No
- ○ Do not know

Permission Predictability: Would you expect this app to request access to your *{permission}*?

- ○ Yes
- ○ No

Permission Sensitivity: When using mobile apps, many people find that there are some resource accesses (permissions) that they are generally comfortable granting, some accesses that they are only comfortable granting under certain conditions, and some accesses are too sensitive that they never or only rarely are comfortable granting. Given the information that this app will request access to your *{permission}*. Please indicate to what extent you agree or disagree with the following statements.

- In general, I do not feel comfortable granting access to my *{permission}*.
- I feel that this app requires access to a very private resource.
- The access to my *{permission}* is very sensitive to me.

Permission Clarity before app interaction:

- I understand the reason for this app to request access to my *{permission}*.
- I have no idea why this app wants access to my *{permission}*.
- It is clear to me why this app needs access to my *{permission}*.

8.1.2 Post-Questionnaire

Now, imagine that you downloaded *{app name}* on your phone to *{objective to use the app}*. Below this text is an interactive mockup app of *{app name}*. Please interact with the app as you would on your own phone until access to your *{permission}* is requested. You can repeat your interaction with the app by clicking the reset button. Then answer the following questions. [Show interactive mockup app same as in Figure 8.1.]

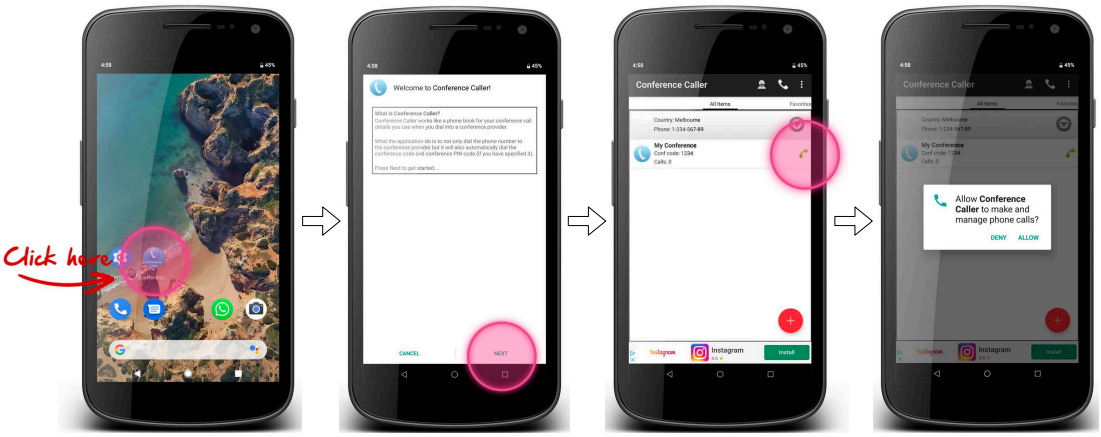


Figure 8.1: Sample interactive mockup app interaction

Decision: Based on your interaction with this app, would you grant this app access to your *{permission}*?

- ○ Yes
- ○ No

Permission Purpose: In your opinion, for what does this app need access to your *{permission}*?

- ○ For the main functionality of the app (app cannot function without it)
- ○ For some additional feature functionality
- ○ Do not know
- ○ For some other reason underline

Permission Clarity after app interaction: After interacting with the above mockup app, please indicate to what extent you agree or disagree with the following statements.

- I understand the reason for this app to request access to my *{permission}*.
- I have no idea why this app wants access to my *{permission}*.
- It is clear to me why this app needs access to my *{permission}*.

Only for requests with rationales: Rationale Origin: Who do you think provided the explanatory message *"This app requires access to your {permission} to..."* that was displayed in a separate dialog immediately before requesting access to your *{permission}*?

- ○ The mobile operating system
- ○ The app developer
- ○ Do not know
- ○ Some other entity underline

Decision Evaluation Scales (DES): In a previous question you chose to *{allow/deny}* this app access to your *{permission}*. We would like to know how you feel about this decision. Please state to what extent you agree or disagree with the following statements.

- **Decision Satisfaction:**
  - I expect to stick with my decision.
  - I am satisfied with my decision.
  - I am doubtful about my choice.
  - I would make the same decision if I had to interact with this app again.
- **Informed Decision:**
  - I am satisfied with the information I received.
  - I know the pros and cons of granting this app access to my *{permission}*.
  - I would have liked more information about how the app will use the access to my *{permission}*.
  - I made a well-informed choice.
- **Decision Control:**
  - I felt pressured by the app to make this decision.
  - The app allowed me to make my own decision.
  - I feel that the app forced me to make this decision.
  - This was my own decision.

Rationale Recall (only for requests with rationales): While interacting with the *{app name}* app you saw a dialog explaining why the app needs access to your *{permission}*. It started with: *"This app requires access to your {permission} to . . . "* Please complete this message as far as you remember. Note: The dialog we are asking you about is the one that immediately preceded the dialog in which you were asked to grant or deny access to your *{permission}*.

[Free-text response]

8.1.3 Demographic Questions

We would like to ask you for some demographic information.

Mobile OS: What operating system are you using on your (primary) mobile phone?

○ Android
○ iOS (iPhone)
○ Windows (Windows Phone)
○ Other underline

Gender: Which gender do you identify most with?

○ Male
○ Female
○ Prefer not to say
○ Other underline

Age: In what year where you born?

[Drop-down list]

Education: What is the highest degree or level of education you have completed?

- ○ Some school, no degree
- ○ High school graduate
- ○ College, no degree
- ○ Bachelor's degree
- ○ Master's degree
- ○ Professional degree
- ○ Doctorate degree

Computer Science Background: Are you studying or have you been working in any of the following areas: information technology, computer science, electronic data processing, electrical engineering, communications technology, or similar?

- ○ Yes
- ○ No

Privacy Concerns:

- Compared to others, I am more sensitive about the way mobile apps handle my personal information.
- To me, it is the most important thing to keep my privacy intact from mobile apps.
- In general, I am very concerned about threats to my personal privacy.

Prior Privacy Experience:

- How often have you personally experienced incidents whereby your personal information was used by some mobile app without your authorization?
- How much have you heard or read during the last year about the use and potential misuse of the information collected from mobile apps?
- How often have you personally been the victim of what you felt was an improper invasion of your privacy from a mobile app?

8.2 Demographics of Participants

Table 8.1 presents the demographic information of our 473 participants.

8.3 Model Fit

We compared all model-building steps using the Akaike Information Criterion (AIC) and likelihood-ratio tests. The model with the lowest AIC was selected as final. For consistency with our study design, variables of interest (step 3) were retained in the DES Control model even if not significant. Table 8.2 reports model fit, marginal R^2, and conditional R^2 for each step across all outcomes.

8.4 User Study Apps

Table 8.3 lists the study apps, their requested permissions, the reasons provided to participants for using each app, and the corresponding rationale messages.

Table 8.1: Demographics of participants

Number of Participants	473		Number of Participants	473	
Computer Science Background					
Yes	176	37.2%			
No	297	62.8%			
Gender			Mobile OS		
Male	296	62.6%	Android	330	69.8%
Female	174	36.8%	iOS	134	28.3%
Other	3	0.6%	Other	9	1.9%
Age			Education		
18–23	20	4.2%	Up to high school	54	11.4%
24–30	128	27.1%	Professional school degree	6	1.3%
31–40	184	38.9%	Some college (no degree)	83	17.5%
41–50	78	16.5%	Bachelor's degree	243	51.4%
51 and over	63	13.3%	Graduate degree	87	18.4%

Table 8.2: Goodness of fit for final models

Decision	AIC	LogLik	Df	P(χ^2)	R^{2m}	R^{2c}
simple regression	2328.14	-1163.07				
step 1: multilevel base (app & user as random effects)	1955.97	-974.98	2	<0.001		0.590
+ step 2: variables from previous work	1487.60	-734.80	6	<0.001	0.462	0.733
+ step 3: variables of interest: timing and rationales	1449.35	-713.68	2	<0.001	0.483	0.765
+ step 4: interaction between timing and rationales	1451.35	-713.68	1	0.986	0.483	0.765
DES Inform						
simple regression	6290.77	-3143.39				
step 1: multilevel base (app & user as random effects)	6013.84	-3002.92	2	<0.001		0.354
+ step 2: variables from previous work & Decision	5746.44	-2862.22	7	<0.001	0.180	0.430
+ step 3: variables of interest: timing and rationales	5647.17	-2810.59	2	<0.001	0.207	0.470
+ step 4: interaction between timing and rationales	5633.44	-2802.72	1	<0.001	0.211	0.476
DES Satis						
simple regression	5500.03	-2748.02				
step 1: multilevel base (app & user as random effects)	4921.63	-2456.82	2	<0.001		0.533
+ step 2: variables from previous work & Decision	4704.05	-2341.02	7	<0.001	0.194	0.544
+ step 3: variables of interest: timing and rationales	4702.12	-2338.06	2	0.052	0.196	0.546
+ step 4: interaction between timing and rationales	4695.43	-2333.72	1	<0.01	0.198	0.549
DES Control						
simple regression	6343.33	-3169.67				
step 1: multilevel base (app & user as random effects)	5350.00	-2671.00	2	<0.001		0.676
+ step 2: variables from previous work & Decision	5245.12	-2611.56	7	<0.001	0.134	0.677
+ step 3: variables of interest: timing and rationales	5243.57	-2608.78	2	0.062	0.136	0.679
+ step 4: interaction between timing and rationales	5243.39	-2607.69	1	0.139	0.136	0.679
Post Clarity						
simple regression	7775.50	-3885.75				
step 1: multilevel base (app & user as random effects)	7401.07	-3696.54	2	<0.001		0.314
+ step 2: variables from previous work	6561.61	-3270.80	6	<0.001	0.470	0.512
+ step 3: variables of interest: timing and rationales	6424.99	-3200.50	2	<0.001	0.502	0.559
+ step 4: interaction between timing and rationales	6418.44	-3196.22	1	<0.01	0.504	0.562

$\mathbf{R^{2m}}$ = Marginal R^2. $\mathbf{R^{2c}}$ = Conditional R^2. $\mathbf{P(\chi^2)}$ = p-value associated with the Chi-squared test.

Table 8.3: User study apps

App	Perm.	Perm. purpose	Goal to use the app *(You want to use this app to ...)*	Rationale message *(This app requires access to your...)*
TextDrive[1]	contacts	visible	block phone calls of some contacts while you're driving.	contacts to display caller names, and block selected contacts.
Conference Caller[1]	phone	main	have a conference call with your work colleagues.	phone to make conference calls.
SContact[1]	phone	hidden	exchange contact information with your business partners.	phone to read device id to uniquely identify your device.
Meteor[2]	location	visible	compare network speed of different locations.	location to show your network accesses on map.
Wifi Time Tracker[2]	location	main	keep track of your working hours.	location to scan for nearby Wi-Fi networks.
AmazeVPN[2]	storage	hidden	use vpn while browsing.	photos, media, and files to manage cache of app data on SD card.
EOS[3]	location	visible	order some delicious sandwiches.	location to find EOS restaurants nearby, and show your location on map.
Cookiegazm[3]	location	main	order food from Cookiegazm.	location to find Cookiegazm restaurants nearby.
Pancakes[3]	storage	hidden	find the next stampede pancake breakfast event.	photos, media, and files to manage cache of app data on SD card.
FaceSwap [4]	mic.	visible	record a video of you and your friend with your faces swapped.	microphone to record face swapped videos with audio.
Beauty Cam[4]	camera	main	take a beautiful selfie.	camera to display stickers on camera view, and take selfies.
Free Fonts for Samsung[4]	phone	hidden	get new fonts for your phone.	phone to read device id to uniquely identify your device.
All Meter[5]	mic.	visible	measure the sound level of your voice.	microphone to measure sound levels in dB.
Loopback[5]	mic.	main	measure the round-trip latency of your voice.	microphone to measure round-trip audio latency.
Tractor Guide[5]	storage	hidden	mark which field areas you have already covered with fertilizer.	photos, media, and files to manage cache of app data on SD card.
Belize Radio World[6]	mic.	visible	record your own channel.	microphone to record your own audio program.
Strobily[6]	camera	main	make your phone's flashlight sync to music.	camera to turn on flashlight.
Cambodian Radio[6]	location	hidden	listen to music.	location for targeted advertisement.
NT Hunting Mate[7]	storage	visible	report an illegal hunting activity.	photos, media, and files to store uncompleted reports.
GoldHunt Free[7]	location	main	find a hidden geocache nearby.	location to show your location on map, and find unfound caches nearby.
Trout Fly Fishing[7]	storage	hidden	learn how to tie a fly.	photos, media, and files to store cache of app data for better performance.
My Weirton[8]	location	visible	report a pothole in Weirton.	location to find reported issues nearby, and show current location on map.
SkyPointer[8]	location	main	find the current position of the ISS in the sky.	location to autocomplete your current location and coordinates.
Monroeville Chamber[8]	storage	hidden	find opening times of the museums in Monroeville.	photos, media, and files to download app content to SD card.
Vehi Care[9]	storage	visible	backup your vehicle's data.	photos, media, and files to store backups of your car data to SD card.
OpenMBTA[9]	location	main	find the closest train station nearby.	location to find train stations nearby, and show your current location on map.
ELCO Chevrolet Cadillac[9]	storage	hidden	buy a used car.	photos, media, and files to download app content to SD card.
Dinosaur Photo Wallpapers[10]	camera	visible	take a selfie with a dinosaur frame.	camera to display dinosaur frames on camera view, and take pictures.
Ice Cream Wallpapers[10]	storage	main	set an ice cream wallpaper as your phone's background.	photos, media, and files to download wallpapers to SD card.
Roses[10]	phone	hidden	send a rose picture to your friend.	phone to read device id to uniquely identify your device.

[1]commun., [2]connection, [3]delivery service, [4]design and art, [5]measurem. tools, [6]music and sound, [7]outdoor activities, [8]places and stars, [9]vehicles and transport, [10]wallpapers

9

Appendix: Rationale Phrasing

9.1 Demographics of Participants

Table 9.1 presents the demographic information of our 960 participants, while Table 9.2 shows their countries of residence.

Table 9.1: Demographics of participants

Number of Participants	960	
Gender		
Male	478	49.8%
Female	466	48.5%
Other	14	1.5%
Prefer not to say	2	0.2%
Age		
18–24	269	28.0%
25–34	416	43.3%
35–44	151	15.7%
45–54	77	8.0%
55–64	36	3.8%
65 and over	11	1.1%
Education		
Less than high school	4	0.4%
High school	133	13.9%
Some college	164	17.1%
Associate degree	61	6.4%
Bachelor degree	421	43.9%
Master degree	141	14.7%
Doctoral degree	14	1.5%
Professional degree	16	1.7%
Other	6	0.6%
Mobile OS		
Android	579	60.3%
iOS	366	38.1%
Windows	15	1.6%

Table 9.2: Residence of participants

Country of Residence	960	
Africa	**119**	**12.4%**
South Africa	119	12.4%
Americas	**358**	**37.3%**
United States	131	13.6%
Mexico	115	12%
Canada	98	10.2%
Chile	14	1.5%
Asia	**10**	**1.0%**
Israel	7	0.7%
Japan	2	0.2%
Korea	1	0.1%
Europe	**433**	**45.1%**
Poland	100	10.4%
Portugal	94	9.8%
Italy	56	5.8%
Spain	37	3.9%
United Kingdom	32	3.3%
Greece	26	2.7%
Hungary	21	2.2%
Estonia	11	1.1%
Latvia	8	0.8%
Netherlands	8	0.8%
Czech Republic	7	0.7%
Finland	6	0.6%
France	5	0.5%
Ireland	4	0.4%
Slovenia	4	0.4%
Switzerland	4	0.4%
Belgium	3	0.3%
Germany	2	0.2%
Sweden	2	0.2%
Austria	1	0.1%
Denmark	1	0.1%
Norway	1	0.1%
Oceania	**40**	**4.2%**
Australia	40	4.2%

9.2 Questionnaire

In this section, you will find the survey questions, keeping in mind that the questions in Section 9.2.1 are repeated for each permission (camera, location, storage, and microphone).

9.2.1 Rationale Questions

On the next pages, you'll find four messages from various smartphone apps. Pretend you've just installed these apps on your phone and the first thing you see upon opening each app is one of these messages. We'd like you to react to the message and then answer a few questions to share your opinion.

Imagine that you just installed the following app on your phone. Please continue to see the next screen of this app. [Show Figure 9.1a]

After opening the app, it shows you the following message. Please take a moment to carefully read this message. [Show Figure 9.1b]

Decision: Would you allow this app access to your *{permission}*? Please carefully read the message in the screenshot before making your decision.

- ○ Allow
- ○ Deny

Decision Satisfaction: In a previous question, you decided to *{allow/deny}* the app access to your *{permission}*. Please indicate your agreement with the following statements concerning your decision: [1–strongly disagree 2–disagree 3–somewhat disagree 4–neither agree nor disagree 5–somewhat agree 6–agree 7–strongly agree]

- I expect to stick with my decision.
- I am satisfied with my decision.
- I am doubtful about my choice.
- I would make the same decision if I had to interact with this app again.
- I am very confident that I made the right decision for myself.

Informed Decision: Please indicate your agreement with the following statements concerning your decision: [1–strongly disagree 2–disagree 3–somewhat disagree 4–neither agree nor disagree 5–somewhat agree 6–agree 7–strongly agree]

- I am satisfied with the information I received.
- I know the pros and cons of granting this app access to my *{permission}*.
- I would have liked more information about how the app will use the access to my *{permission}*.
- I made a well-informed choice.
- I know exactly why the app needs access to my *{permission}*.

Decision Control: Please indicate your agreement with the following statements concerning your decision: [1–strongly disagree 2–disagree 3–somewhat disagree 4–neither agree nor disagree 5–somewhat agree 6–agree 7–strongly agree]

- I felt pressured by the app to make this decision.
- The app allowed me to make my own decision.
- I feel that the app forced me to make this decision.
- This was my own decision.
- I felt that the app would exclude me from using its features if I refused to grant access to my *{permission}*.

(a) Home screen **(b)** Rationale screen

Figure 9.1: Sample screens from the questionnaire.

9.2.2 Demographic Questions

Gender: Which gender do you identify most with?

- ○ Male
- ○ Female
- ○ Prefer not to say
- ○ Other _____

Year of Birth: What is your year of birth? _____
Education: What is the highest level of school you have completed or the highest degree you have received?

- ○ Less than high school degree
- ○ High school graduate (high school diploma or equivalent including GED)
- ○ Some college but no degree
- ○ Associate degree in college (2–year)
- ○ Bachelor's degree in college (4–year)
- ○ Master's degree
- ○ Doctoral degree
- ○ Professional degree (JD, MD)
- ○ Something else, namely: _____

Mobile OS: What operating system are you using on your (primary) mobile phone?

- ○ Android
- ○ iOS (iPhone)
- ○ Windows (Windows Phone)
- ○ Other _____

Privacy Concerns: Please rate the following statements: [1–strongly disagree 2–disagree 3–somewhat disagree 4–neither agree nor disagree 5–somewhat agree 6–agree 7–strongly agree]

- Compared to others, I am more sensitive about the way mobile apps handle my personal information.
- To me, it is the most important thing to keep my privacy intact from mobile apps.
- In general, I am very concerned about threats to my personal privacy.

Prior Privacy Experience: Please answer the following questions: [1–never 2–very rarely 3–rarely 4–occasionally 5–frequently 6–very frequently 7–always]

- How often have you personally experienced incidents whereby your personal information was used by some mobile app without your authorization?
- How much have you heard or read during the last year about the use and potential misuse of the information collected from mobile apps?
- How often have you personally been the victim of what you felt was an improper invasion of your privacy from a mobile app?

Use My Answers: Sometimes, when people take part in a survey, they may not pay full attention or get distracted. When this happens, the answers they provide may not be good for scientific research. Please answer honestly: Were you diligent and attentive when answering this questionnaire? You will be paid for your participation regardless of your answer.

○ No
○ Yes

9.3 DES Item Fit and Consistency

We calculated multilevel internal consistency using McDonald's Omega [56] and determined the factor loadings for all items in the Decision Evaluation Scales (DES). Table 9.3 presents the results for all items and subscales of the DES. The newly included Informed Decision and Decision Satisfaction items showed a very good fit. In contrast, the item added to Decision Control had a somewhat weaker fit but was still considered acceptable. Despite variations in factor loadings, all subscales demonstrated good internal consistency within and across participants, indicating that the scales' consistency remained robust.

9.4 Model Fit

We statistically compared all steps in the model-building process using the akaike information criterion (AIC) and the likelihood-ratio tests. The model that described our data best and had the lowest AIC score was selected as the final model. Table 9.4 presents the goodness of fit, marginal R^2, and conditional R^2 for each step in the model-building process of all outcome variables.

Table 9.3: Standardized item fit and internal consistency measures for the DES subscales.

	DES Inform	DES Satis	DES Control
I am satisfied with the information I received.	0.87		
I know the pros and cons of granting this app access to my {permission}.	0.58		
I would have liked more information about how the app will use the access to my {permission}.	0.60		
I made a well-informed choice.	0.69		
I know exactly why the app needs access to my {permission}.	0.82		
I expect to stick with my decision.		0.81	
I am satisfied with my decision.		0.88	
I am doubtful about my choice.		0.72	
I would make the same decision if I had to interact with this app again.		0.80	
I am very confident that I made the right decision for myself.		0.88	
I felt pressured by the app to make this decision.			0.85
The app allowed me to make my own decision.			0.49
I feel that the app forced me to make this decision.			0.88
This was my own decision.			0.45
I felt that the app would exclude me from using its features if I refused to grant access to my {permission}.			0.45
$\omega_{between}$	0.84	0.93	0.84
ω_{within}	0.87	0.89	0.77

Table 9.4: Goodness of fit for final models

Decision	AIC	LogLik	Df	P(χ^2)	R^{2m}	R^{2c}
simple regression	5050.1	-2524.0				
step 1: multilevel base (user & perm. as random effects)	4963.5	-2478.8	2	<0.001		0.161
+ step 2: variables from previous work	4882.5	-2436.3	2	<0.001	0.041	0.167
+ step 3: variables of interest: rationale building blocks	4874.3	-2423.1	9	0.002	0.050	0.177
+ step 4: interactions between building blocks	4905.6	-2411.8	27	0.702	0.058	0.188
DES Inform						
simple regression	13661.7	-6828.8				
step 1: multilevel base (user & perm. as random effects)	13252.7	-6622.4	2	<0.001		0.299
+ step 2: variables from previous work & Decision	12642.9	-6314.4	3	<0.001	0.133	0.414
+ step 3: variables of interest: rationale building blocks	12621.0	-6294.5	9	<0.001	0.142	0.413
+ step 4: interactions between building blocks	12667.4	-6290.7	27	0.999	0.143	0.414
DES Satis						
simple regression	11489.1	-5742.5				
step 1: multilevel base (user & perm. as random effects)	10903.8	-5447.9	2	<0.001		0.364
+ step 2: variables from previous work & Decision	10800.1	-5393.0	3	<0.001	0.033	0.377
+ step 3: variables of interest: rationale building blocks	10792.5	-5380.2	9	0.002	0.037	0.381
+ step 4: interactions between building blocks	10828.0	-5371.0	27	0.889	0.041	0.384
DES Control						
simple regression	12128.8	-6062.4				
step 1: multilevel base (user & perm. as random effects)	11036.5	-5514.2	2	<0.001		0.496
+ step 2: variables from previous work & Decision	10990.7	-5488.4	3	<0.001	0.025	0.496
+ step 3: variables of interest: rationale building blocks	10982.7	-5475.4	9	0.002	0.029	0.499
+ step 4: interactions between building blocks	11017.0	-5465.5	27	0.844	0.032	0.503

$\mathbf{R^{2m}}$ = Marginal R^2. $\mathbf{R^{2c}}$ = Conditional R^2. $\mathbf{P(\chi^2)}$ = p-value associated with the Chi-squared test.

10
Appendix: Web Rationales

10.1 Crawler

10.1.1 Contributon of DOM Interactions

Our crawler, described in Section 6.3.1, interacts with webpages by clicking on elements likely to trigger permission-related functionalities. This interaction is guided by a manually curated list of heuristics, which we created by reviewing the source code of 500 random sites with permission prompts. Particularly, we checked if the occurrence of prompts on these sites relied on user interaction and identified the relevant DOM selectors (e.g., node ID attribute, class name, etc). Then, we grouped these selectors based on their similarity and created a set of heuristics to guide the crawler on which elements to click. Table 10.2 presents the complete list of heuristics.

We conducted additional experiments to quantify the contribution of our interactive crawler in triggering permission prompts and APIs. To do that, we randomly selected 100 URLs from the 770K seed URLs that use any permission concept, and an additional 100 URLs from each of the four permission concepts to ensure that there are sufficient samples from each permission type, resulting in 500 URLs. We then compared the number of observed API calls or prompts with and without crawler page interactions. To increase the confidence in results, we repeated the random sampling and our experiment two times, testing a total of 1K webpages, and take the aggregated result.

In total, we observed that incorporating page interaction heuristics more than doubled the likelihood of encountering browser prompts (and thereby rationales) at runtime, increasing observed calls to permission-gated APIs from 5.1% to 10.4% of webpages. Table 10.1 summarizes our experimental results.

Table 10.1: Contribution of crawler DOM interactions in triggering permission API calls or prompts based on heuristics in Table 10.2. The left part shows the percentage of pages with captured API calls, whereas the right part shows the absolute number of pages with observed calls for individual permissions. Legend: S_i represents the random subset i of the dataset.

Crawler	Experiment	Pages	Calls	Observ.	Cam.	Mic.	Geo.	Notif.
Baseline	Run #1	S_1: 500	23	4.6%	3	1	2	17
	Run #2	S_2: 500	28	5.6%	0	0	11	17
	Total	1,000	51	**5.1%**	3	1	13	34
Interactive	Run #1	S_1: 500	54	10.8%	0	3	21	30
	Run #2	S_2: 500	50	10%	0	3	19	28
	Total	1,000	104	**10.4%**	0	6	40	58

10.1.2 Understanding Prompt Detection Challenges

Our automated crawler observed permission prompts on only ∼20% of the pages from the seed list (all URLs on the seed list use at least one of the most popular permission-gated web API according to Chrome telemetry). To understand the underlying causes, we randomly selected 100 webpages where the crawler missed permission prompts and manually investigated the reasons.

Table 10.2: The complete list of node selector heuristics the crawler uses for page clicks to trigger permission prompts.

Permission	DOM Selector

Notification	`[id=onesignal-slidedown-allow-button]`
	`button[class*="cleverpush-confirm-btn-allow"]`
	`button[class*="dn-slide-accept-btn"]`
	`button[class*="js-pushowl-yes-button"]`
	`//button[contains(.,"notification")]`
	`[data-test-id="push-subscription-cta-accept"]`
	`div[id="btn-allow"]`
	`'div[class*="btn-notification"]`
	`//div[text()="Zulassen"]`
	`//div[text()="allow"]`
	`a[class*="allow"]`
	`[class*="allow"]`
	`[id*="allow"]`
	`[id=push-popup-yes]`
	`[class*="approve"]`
	`[class*="btn-notification"]`

Geolocation	`[data-qa-id="use-my-location-btn"]`
	`[class*="location-btn"]`
	`getElementsByTagName("m-locate-me")`
	`[class*=js-location-button]`
	`[class*=location]`
	`[id*=location]`

Camera	`[id*=video]`
	`[class*=allow-camera]`
	`[class*=enable-camera]`
	`[class*=use-my-camera]`
	`[class*=use-camera]`
	`[class*=camera]`
	`//p[contains(.,"Use my camera")]`
	`//button[contains(.,"Get started now")]`

Microphone	`[id*=microphone]`
	`[class*=microphone]`
	`[class*=soundcheck]`
	`[class*=tuneron]`
	`[class*=input__voice-search]`
	`[title*=speech-to-text]`
	`[class*=speech-to-text]`
	`[class*=voice]`
	`[class*=btn-record]`
	`[class*=music-box__buttons__button]`

We found that 73% of the cases were due to common crawling challenges: reaching deep application states (28%), handling DOM interactions (20%), authentication barriers (14%), and bot detection mechanisms (10%). In 11% of the cases, the target webpages were no longer active. Additionally, 16% of the URLs from the telemetry dataset were sanitized for privacy, leading to discrepancies between the pages our crawler visited and the actual pages where permissions were observed. Table 10.3 summarizes our findings.

Table 10.3: Distribution of reasons automated crawling missed permission prompts on a sample of 100 sites.

#	Reason	Count
1	Complex Application State	28
2	DOM Interaction Required	20
3	URL Sanitized	16
4	Authentication Required	14
5	Geoblocked Access	12
6	Page Inactive	11
7	Captcha/Bot Prevention	10

10.2 Role of Prompts in Rationale Identification

We conducted two experiments to explore the presence and location of rationales alongside web permission prompts. In the first, we manually analyzed 100 randomly selected pages from the dataset where the crawler detected prompts. Among these, only 10 contained a discernible rationale, whether in text, UI elements, or both, accounting for 10% of the cases, of which almost half (i.e., 4.6%) were purely based on English text. Extrapolating this finding to the whole dataset of 162K pages with observed prompts suggests a noteworthy scarcity of rationales provided alongside permission prompts, amounting to approximately 16.2K pages in total, of which 7.4K are expected to contain rationales based on English text.

In the second experiment, we analyzed 1K random pages identified as having prompts. We created accounts and logged in to assess rationales after login, although in rare cases, this was not feasible due to account requirements. Our analysis revealed rationales on 113 sites, with 19 of these found after login. This indicates that approximately 17% of the rationales are post-authentication. We observed that the total number of rationales discovered (113 out of 1K) is close to the 10% found in the first experiment, suggesting consistency across both experiments.

10.3 Experience Sampling Questionnaire

Figure 10.1 depicts how the experience sampling questionnaire appeared in Chrome. Please also see [71] for additional details on this method. The variable *$capability* can take the value *"geolocation"*, *"camera"*, or *"microphone"*. The following are the corresponding questions.

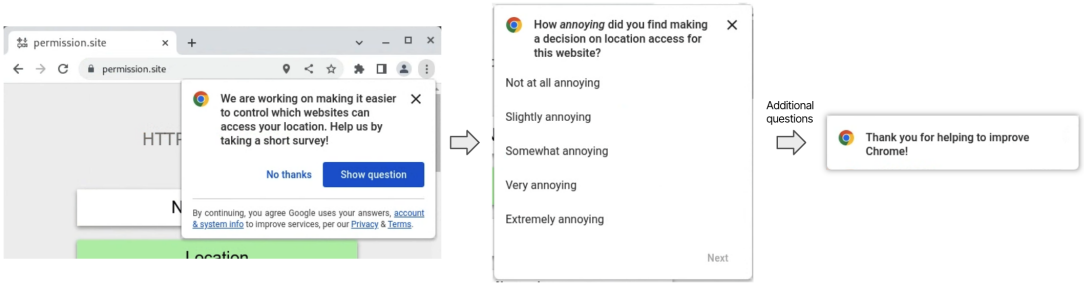


Figure 10.1: Screenshot of questionnaire invitation and subsequent screens. (71)

Q0: A website just asked for access to your *$capability*. Help us improve how websites ask for access by taking this 1-minute survey!

Q1: [not shared with us]

Q2: How annoying did you find having to make a decision on *$capability* access for this website?

- ○ Not at all annoying
- ○ Slightly annoying
- ○ Somewhat annoying
- ○ Very annoying
- ○ Extremely annoying

Q3: How easy or difficult did you find making a decision on *$capability* access for this website?

- ○ Very difficult
- ○ Somewhat difficult
- ○ Neither difficult nor easy
- ○ Somewhat easy
- ○ Very easy

Q4: [not shared with us]

Q5: Thank you for helping to improve Chrome!

10.4 LLM Filtering Prompt

We used the following few-shot prompt to identify text snippets related to permission concepts. The few-shot examples are from real websites.

10.5 Library Detection Rules

Table 10.4 provides a summary of the library detection rules derived from the permission rationales in our rationale catalog. We used these rules to mine similar patterns and

LLM Rationale Filtering Prompt

You are an assistant trying to help users manage their browsers. You are given a sentence and you have to decide if it is a rationale or not. The definition of rationale is the following: a rationale is a sentence from a website that asks (directly or indirectly) a user to allow access to one of the following devices: webcam, push notifications, microphone, user's location. If you decide that the sentence is a rationale, you have to write the name of the relevant device.

Sentence: Get Breaking News Alerts. We'll send you latest news updates through the day. You can manage them any time from your browser settings.
Answer: notifications

Sentence: However, you are not logged in. Log in or Sign up to receive price alerts.
Answer: No

Sentence: error we did not manage to get access to your location
Answer: location

Sentence: this is a very good model, it can record audio and video
Answer: No

Sentence: In order to reliably test your equipment, this page requires your browser's permission to detect your webcam and microphone.
Answer: webcam, microphone

Sentence: Look at streamers on cam!
Answer: No

Sentence: You need to connect a microphone.
Answer: microphone

Sentence: You can find us at the following address: 1234 Main St, Anytown, CA, 12345
Answer: No

Sentence: [TEXT_PLACEHOLDER]
Answer: ?

identify additional uses of these libraries within our broader dataset, which consists of snapshots of hundreds of thousands of webpages collected during our web crawling.

False Positives of Library Detection Rules. We designed our library detection rules to be strict, minimizing the chance of false positives even at the expense of potential false negatives. To assess the false positive rate, we randomly selected three instances hit by each of the 32 signatures and manually vetted if the hit was a false positive. The results confirmed that our detection rules are robust, with all reviewed cases being true positives. This finding was expected, as the majority of the rules test for the presence of specific and relatively long string identifiers within HTML tag attributes, significantly reducing the likelihood of collisions.

Machine Learning vs. Library Signatures. We observed that signature-based rationale detection method, which operates directly on HTML code, excels at capturing

instances where rationale text may not be immediately visible or requires user interaction to load, though compiling and maintaining a comprehensive list of signatures can be challenging. Conversely, the ML-based approach targets text within the rendered DOM, detecting both library and custom rationales, including libraries not found via the signature matching approach due to missing signatures in webpages, but may miss cases where text is split or not fully loaded, underscoring the complementary nature of these techniques.

Table 10.4: Summary of library detection rules that we extracted from web permission rationales. The rules are based on the Beautiful Soup HTML parser (134).

Library	**Rule**
iZooto	doc.find_all(attrs='class':'iz-news-hub-noti-blockd-txt') doc.find(attrs='id':'enable-turned-off-notis-cta')
OneSignal	doc.find_all(attrs='class':'modal-dialog') doc.find_all(attrs='class':'modal-notify') doc.find_all(attrs='class':'modal-body-message') doc.find(attrs='id':'onesignal-slidedown-container')
PushEngage	doc.find(attrs='id':'pe-widget-bell-launcher-message')
Smart Push	doc.find(attrs='id':'smart_push_smio_msg') doc.find(attrs='id':'smart_push_smio_note') doc.find(attrs='id':'smart_push_smio_not_allow') doc.find(attrs='id':'smart_push_smio_allow') doc.find(attrs='id':'smart_push_smio_footer') doc.find(attrs='id':'smart_push_arrow_bottom') doc.find(attrs='id':'smart_push_smio_agreement_contents') doc.find(attrs='id':'smart_push_smio_agreement_option') doc.find(attrs='id':'smart_push_gdpr_icon_message')" doc.find(attrs='id':'smart_push_smio_note') doc.find(attrs='id':'smart_push_smio_not_allow') doc.find(attrs='id':'smart_push_smio_allow')
Moe-push	doc.find(attrs='id':'moe-push-div')
PushOWL	doc.find(attrs='id':'pushowl-simple-toast-content') doc.find_all(attrs='class':'pushowl-simple-toast')
Perfecty	doc.find(attrs='id':'perfecty-push-settings-subscribed') doc.find(attrs='id':'perfecty-push-dialog-container')
Webpushr	doc.find_all(attrs='class':'webpushr-bell-theme-dark') doc.find_all(attrs='class':'webpushr-toggle-bell-popup')
Superstore-finder-wp	doc.find(attrs='id':'storeLocator___mapStatus___inner') doc.find(attrs='id':'storeLocator___mapStatus___closer')
Storerocket	doc.find_all(attrs='class':'storerocket-lead') doc.find_all(attrs='class':'storerocket-message-list') doc.find_all(attrs='class':'storerocket-initial-message-content') doc.find_all(attrs='class':'storerocket-error')
Total Rules	32

10.6 Rationale Clustering and Examples

We used agglomerative clustering in Section 6.3.3 to group together similar rationale embeddings generated via all-MiniLM-L6-v2 [46] sentence transformer. We chose the

agglomerative clustering algorithm due to its capability to merge data points based on proximity measures, thereby facilitating the identification of semantic relations and keyword occurrences within the rationale texts. Our implementation uses the clustering model from the sklearn library [139], setting the affinity parameter to Euclidean distance and using a distance threshold of 3.5 to ensure precise and meaningful clusters. Table 10.5 presents examples of rationale texts from each of the 70 clusters identified in Section 6.6.1 following the above clustering methodology.

Table 10.5: Examples of rationales from each of the 70 sentence transformer subclusters. IDs in the table represent cluster names and are composed of the first letter of the permission name followed by a group identifier. For example, G0 stands for Geolocation0 subcluster and belongs to the Geolocation cluster. **Legend:** G = Geolocation. N = Notification. M = Microphone. C = Camera. CM = Camera_Microphone. CMG = Camera_Microphone_Geolocation.

ID	Rationale	Domain
G0	It is mandatory to allow location of your browser to open an account through Video KYC	onlinesb.pnbindia.in
G1	To order online, please use the store locator below. To save your location for future online orders, press the “Set My Location” button on the location you are ordering from.	picklemans.com
G2	Requesting location access...	creedboutique.com
G3	Step 2: Click on “Location” in the options presented and then choose “Share live location”.	imyfone.com
G4	Click Allow to easily find a bank and be in the know for all bank information!	allusbanks.com
G5	Click map to set your location	navigateme.lincoln.ac.uk
G6	We’re searching for local stores. Your browser may ask for permission to use your location. Click "Allow" to sort the search results by distance.	theroomplace.com
G7	Please Allow GPS So That App Features May Be Enabled. Please Enable Location Service for Browser, and Clear Browser History Before Retry	app.masa.plus
G8	Geolocation Information. We may request access or permission	test2fly.carekore.app
G9	Your location is not permitted	rctiplus.com
G10	Click Allow for all Jet’s Pizza menu updates and find a location near You!	menuwithprice.com
G11	Enter your address or zip code in the search bar below, adjust your search radius in the dropdown on the right, and click search. You may also click the arrow to geolocate and search from your current location.	rotech.com
G12	Click Now to find available Free Dental Clinic in your area.	livefit101.com
G13	Use your current location or enter search criteria in the form. Then choose a search radius and select the Search button to find dealers in your area.	windsorwindows.com
G14	Please turn on your location setting for your browser to see your nearest store. Alternatively, you can search by entering your city/postcode above or simply browse the map below.	charlestyrwhitt.com
G15	Allow the browser to use your location. Use current location	grubhub.com
G16	Click Accept and an initial pop-up will appear on your screen. Click “Continue” to go to your device’s native Permission For Tracking pop-up.	playtikaprod.service-now.com
G17	Your location could not be determined. Click here to use your current location or enter your zip code in form above.	centier.com

ID	Rationale	Domain
G18	Allow us to access your location. We need your location to provide you with the best experience. Your location is safe with us. Allow Location	ajio.com
G19	Please enable your browser to allow this site to use your location	deltadentalnc.com
G20	Click Allow to get more free information about Public Housing Waiting List!	uslowcosthousing.com
M0	Once the number is entered, simply click on the “Call” button on the bottom of the dialpad. You will be prompted to allow PopTox to access your mic. Click on “Allow” for us to connect your call. Make sure to not “Deny” mic permission.	poptox.com
M1	To identify your range we will need to use your microphone.	singingcarrots.com
M2	Voice To Text Converter Click on the microphone icon and begin speaking for as long as you like.	unicodeconverter.info
M3	Your camera access is blocked. We can’t continue without video. To connect with sign language support, allow access to your camera and microphone. Allow Access	signtime.apple
M4	If you are prompted, click to Allow access to the microphone.	htsdl.com
M5	Click here to test your mic.	xujenna.com
M6	To record audio messages, you must allow access to the microphone. I have authorized access, try again.	donationalerts.com
M7	You’ll get a pop up from your browser asking to allow to use the microphone. Click to allow, so the violin tuner can pick up the note you’re playing and tell you if it’s in tune.	violinlounge.com
M8	This is a simple online microphone test so you can check whether your microphone works correctly. It’s great before you start a Zoom call or any other video or audio-only call that requires a working microphone to be connected to your desktop or laptop computer. To begin the mic test, simply click the ’Start Test’ button above.	test-microphone.com
M9	The microphone is not connected	micworker.com
M10	You will be asked to provide access to your microphone. App does not send any audio stream data to the servers.	bpmtech.no
N0	Get notified when you move less. The reminders function will ensure you are always on track with your health goals.	reliancedigital.in
N1	You are advised to subscribe with sarvgyan to receive all latest updates & notification for these & other exams.	sarvgyan.com
N2	Don’t forget to subscribe to receive notifications of our new free recipes.	patterns.xn–amgurum-sfb.com
N3	Allow notification permission and refresh this page.	alerts.tbsnews.net
N4	Sign up to receive updates	bata.com.pk
N5	Get a notification when price drops below Rs.699.00 PKR.	jobsearch.childrens.com
N6	Get notified about opportunities that may interest you.	hannity.com
N7	Don’t miss out on important news! Click ’Allow’ for informative articles and updates.	unifi.com.my
N8	Looking to boost your credit? Allow updates to receive personalized alerts	creditcardsearching.thedimepress.com
N9	Click Allow to stay updated with all DMV practice tests!	dmv-test-pro.com
N10	Allow your browser to receive notifications	rainbowloom.de
N11	Sign Up for Alerts. Receive alerts from Berkeley County	berkeleycountysc.gov
N12	Join our notification feed if you want to get the latest Movies, TV Series, Exciting updated Content, and Many More!!!	sunplex.net

ID	Rationale	Domain
N13	Get notified about ride updates & discounts For example: "Your Lyft driver is here!" or "Get $5 in credit" Notifications are blocked. Please follow these instructions to allow this site to show notifications.	ride.lyft.com
N14	Allow alternet.org to send web push notifications to your desktop.	alternet.org
N15	Click Allow for all latest coupons and discounts for Vistaprint!	coupon.hoursguide.com
N16	With your subscription, you'll get email alerts and push notifications to keep you up to speed on the action.	tradersmith.in
N17	Subscribe to our push notifications. No Thanks Allow	in.tubecorporate.com
N18	You have blocked receiving notifications from https://www.lostiempos.com. Please change the browser site settings in order to receive notification	www-lostiempos-com.gravitec.net
N19	So don't wait. Fill out our form to request the loan you've been searching for with Quick Loans. Stay updated on your loan! Click 'Allow' to ensure you receive important updates	quickloans.cash
N20	marionetka.com Would like to send you notifications: Allow, Discard	marionetka.com
N21	Stay up-to-date with SET News	careers360.com
N22	To receive notification from SmartThings Find, you must turn on the notification under settings.	samsung.com
N23	www.zeberka.pl would like to send you web push notifications. These may include commercial information regarding special offers and discount coupons on its own behalf and on behalf of its co-operators. To opt out, turn off notifications from www.zeberka.pl Turning off notifications will be possible at any moment, by clicking the button below.	relaxandwax.com
N24	No locations found near you, but we'd love to change that. Get notified when we add a location nearby Notify Me Reset Search Oops! Something went wrong. This page didn't load Google Maps correctly. See the JavaScript console for technical details.	cashify.in
N25	Disable notifications for WhatsApp to go Invisible On WhatsApp	samsung.com
N26	Don't miss out on best offers! Allow us to send you awesome updates and offers! Don't Allow	sarkariyojnaa.com
N27	CIO wants to show you notifications	cio.com
C0	Click Allow for all latest tricky DMV road sign tests!	flirt4free.com
C1	Under Camera, select "Allow" or "Ask".	readypay.co
C2	Hit the SCAN NOW button to launch the in-browser scanner. You may be prompted for camera access.	coomeet.me
C3	Activate your camera and start chatting. Video chat applications are a fun means to meet all different sorts of people from all over the globe.	echat.live
C4	Turn on the camera permission in your browser to continue further	qrcodescanneronline.com
C5	Use your Camera to start VideoChat	veed.io
C6	After allowed camera permission, just focus device camera to the WiFi QR Code and this tool will scan WiFi QR Code immediately.	megavirt.com
CM0	Give us access to this device, if You have to make free video calls.	globfone.com
CM1	You will need to allow access to your camera and microphone for the video consultation. You can use any computer with a webcam and microphone enabled or a smartphone with a camera.	essential.doxy.me
CMG0	Give access to your webcam, mic, and location if required. Click "Allow" where necessary.	omeglealternative.com