Article

new media & society
1–17

DOI: 10.1177/14614448241270541
journals.sagepub.com/home/nms


"Track every move": Analyzing developers' privacy discourse in GitHub README files

Keren Levi-Eshkol
University of Haifa, Israel

Rivka Ribak
University of Haifa, Israel

Abstract

We adopt a socio-material perspective to examine how developers translate privacy, as a social value, into user applications. Our comprehensive survey of the research on developers' privacy highlights their key position as privacy mediators and their forums as productive settings for unobtrusive studies of their discourse. The open-source code-sharing platform GitHub contains both discourse and code; by focusing on GitHub, we analyzed nearly 60,000 README files created between 2008 and 2020 that include the term "privacy," studying quantitatively and qualitatively how discourse is translated into code. Using VOSviewer.com, we identified two main word clusters: "security" and "privacy policy." Voyant-tools.org confirmed these findings, suggesting that some references elaborate on practices that safeguard privacy, while others discuss policy as a means of complying with both public and, ironically, commercial regulations. A closer reading of the files reveals that even privacy enthusiasts may inadvertently promote code that poses threats to privacy.

Keywords

Developers, GitHub, hybrid method, privacy by design, privacy policy, README files

Corresponding author:
Rivka Ribak, Department of Communication, University of Haifa, Haifa 3498838, Israel.
Email: rribak@com.haifa.ac.il

Introduction

Personal information is an essential commodity in the digital marketplace. As more products become an exchange of information *about* the user for information *for* the user—for example, user location for navigation and weather forecast, past purchases for relevant recommendations—product managers and developers, who are responsible for the software design, have become important factors in the decision what personal information will be collected, where it will be stored, how it will be protected, and whether it will be shared with other commercial entities. In this sense, developers may be seen as mediators (Frenkel, 2005) whose daily work involves translating social norms and values into material products.

Despite this key position, developers' coding decisions and practices seem to remain obscure. The code they write is often described as a technical patchwork (Bucher, 2016; Kitchin, 2017), which is proprietary and not knowable (Pasquale, 2015). In recent years, however, a growing body of research has attempted to gain insight into these intricate processes of mediation, using a variety of methods such as ethnography (e.g. Waldman, 2021), developer assignments (e.g. Li et al., 2022), interviews (e.g. Peixoto et al., 2020), surveys (e.g. Spiekermann et al., 2018), and discourse analysis (e.g. Shilton and Greene, 2019). This article aims to contribute to the understanding of developers as privacy mediators by studying their notions of privacy closest to where and when they are materialized in code. Specifically, by analyzing how developers describe the code they write on the open-source code-sharing platform GitHub, we illuminate both the processes of mediation—how social values become inscribed in digital products—and the particular tensions involved in translating privacy into code in an industry driven by data extraction and accumulation.

The questions we are interested in, *how developers perceive privacy and how they practice it*, draw on three intellectual traditions: They are embedded in digital materialism (Casemajor, 2015; Reichert and Richterich, 2015), which suggests that social ideas acquire material form in interfaces, programs, and applications. In addition, they draw on Values In Design (VID; Friedman, 1996; Nissenbaum, 2005), a more practice-oriented tradition that studies products, beliefs, and work habits of engineers in order to improve design both ethically and practically. Finally and more fundamentally, our questions are inspired by developer studies (Coleman, 2010; Kotliar, 2020; Takhteyev, 2012; Wajcman, 2019) that pay close attention to the ecosystems and the predicaments in which hackers, developers, programmers, product designers, data scientists, or engineers write and use code. As a code-sharing platform, GitHub allows us to explore these questions since here developers share code that materializes in digital products; here they mediate social values into that code; and here they act as members of developers' community of practice.

Materializing privacy

Thinkers from Ellul to Kittler and from Latour to Winner demonstrate the significance of technology for any understanding and conceptualization of communication (Peters, 2017). Their works cast into doubt the immateriality assigned to digital culture (Passoth, 2019), pointing out that "the digital information has a material substrate, and that the

trope of immateriality obscures the labor, materials and natural resources of digital culture" (Casemajor, 2015: 13). However, while materialism as a theoretical framework assumes that all things in the world are rooted in matter, the digitization of matter requires an interpretation that takes into account its intangibility. Digital materialistic research, therefore, addresses the components that are part of and the processes that occur before, during, and after the use of digital products (Reichert and Richterich, 2015). This perspective sheds light on the interactions between human and non-human actors involved in the production and consumption of media (Parikka, 2015), distinguishing between three dimensions of digital matter: the physical—namely the hardware; the logical—the software; and the conceptual—the functional product obtained from the digital display (Hiller, 2015). Thus, the study of the "product" must involve not only the way in which it appears on the screen, but also its ongoing logical development and the physical dynamics that occur before and throughout its translation into functional meaning.

Furthermore, developers inscribe social values and practices into the code they produce, thereby coloring physical and logical materiality with cultural specificity. In turn, their products afford their users with particular properties, parameters, and procedures. Studies that analyze the social structure and meaning of software design (Gottlieb and Cornet, 2021; Poulsen and Kvåle, 2018 Zhao et al., 2014) underscore not only the material aspects of these technologies, but also the ethical considerations that are implicated in their production, and their future human and environmental impact. Thus, by shifting our attention to the ways in which "privacy" is produced in developers' discourse and code projects, digital materialism, VID and developer studies invite us to study online privacy as a socio-material concept and as a knowable practice.

Studying developers' privacy

The thickest description of developers' privacy practices comes from ethnographic accounts. In that tradition, Waldman (2021) participated in development meetings in three companies, attended industry and expert conferences, studied trade and legal documents, conducted interviews with 125 industry workers, and more. His 4-year bottom-up study shows how those involved adopt unquestioningly the neoliberal rationale of the industry's modus operandi, based on a conception of data as "the new oil" (p. 63) within informational capitalism. His informants viewed regulation as an impediment to the industry's creativity. They believed that the users divulged personal information of their own free will, and at the same time, that the companies were safeguarding users' privacy—views shared by 21 Google and Facebook workers (Jørgensen, 2018). Drawing on interviews, visits, and industry events, Jørgensen found that workers in these companies rationalized the ways in which their employers violated users' human rights: "I don't see a conflict between the business model and privacy, provided individual users are in control" (p. 345). Shilton's work suggests that this disconnect originates in developers' education (Boyd and Shilton, 2021). As an advocate for social values and ethnographer of design, she conducted participant observation in a computer science lab and analyzed over 200 hours of field notes and more than 30 interviews, taking into account her changing positioning in the lab as well (Shilton, 2013). Shilton's analysis identifies seven "values levers"—"activities that pried open conversations about anti-surveillance values

in the midst of the rush toward software products and publications" (p. 380), for example, when engineers worked in interdisciplinary teams (particularly ones not limited to computer science majors), or were forced to have their values articulated when they wished to get funding.

Clearly, ethnography provides the richest, most thorough account of developers' work, and its observations allow for perceptive, nuanced analyses of its deep plays (Geertz, 2005). However, by its very essence, it is time-consuming in a field that is rapidly evolving, and requires entry into walled ecosystems that are increasingly proprietary (Kitchin, 2017; Pasquale, 2015). In order to overcome these difficulties, some researchers focus on individual developers' notions of privacy, using interviews that are occasionally complemented by surveys of larger samples (Alomar and Egelman, 2022; Balebako et al., 2014; Bednar et al., 2019; Canedo et al., 2023; Hadar et al., 2018; Peixoto et al., 2020; Ribak, 2019; Tahaei et al., 2021). The interviews paint a picture of overworked developers engaged with their particular projects, who are largely uninterested in privacy, who do not seek assistance upon finding privacy guidelines complicated and difficult to apply, and who use ready-made code whose privacy violations are beyond them. Even Bednar et al. (2019), who conducted interviews with four senior systems engineers and two heads of academic software groups which they estimate to have amassed more than 60 years of experience, report that the incorporation of privacy mechanisms was "inconvenient" and "challenging" (p. 129). They found that company lawyers were uninformed and unreasonable ("it was a nightmare to explain to her certain things and also to know from her the regulations," p. 135), and hesitated over their responsibility to safeguard users' privacy. In an interesting twist, the challenges mentioned in interviews with 12 software teams' "privacy champions" reverberate these tendencies. The advocates reported that they face developers' indifference, tensions between priorities, lack of standardization and evaluation metrics, technical complexity, and communication issues between stakeholders, including ones related to diversity ("Sometimes men are like, 'Why would you need to protect a phone number more?'") (Tahaei et al., 2021: 7).

The exceptional sample composed of "privacy champions" highlights the tendency of most of the interview studies to ask representative developers to reflect upon issues they do not normally consider, to verbalize their work practices in hindsight, and to justify their beliefs and behaviors in the face of interviewers who may value what they themselves possibly do not. Thus, a complementary method presents developers with an assignment that involves a potential threat to privacy, focusing on their reasoning and the actual solutions they provide (e.g. Acar et al., 2016; van der Linden et al., 2020; Wong et al., 2017). This rationale—"rather than asking what [developers] think they would do, or what they think they did in the past, participants are asked about what they just did" (Senarath and Arachchilage, 2018: 1850)—informs a series of studies that simulate actual design environments to explore privacy decision-making as it happens. For instance, Tahaei et al. (2022b) asked developers to "think aloud" as they integrated a banner ad that would be compliant with privacy regulations in a provided app. They found that ad network documentation for privacy regulations is poorly designed, that developers find it challenging to locate relevant information, and that building and applying all the regulations may be even more complicated.

Similarly, Li et al. (2022) observed and interviewed 12 iOS developers in their attempts to create accurately and efficiently "privacy nutrition labels" (required by Apple). Their detailed analysis calls attention, in particular, to the "unknown unknowns"—those situations in which developers are unaware of the errors they have introduced into the privacy labels they prepare.

Such accountability interventions (Boyd, 2021), values elicitation tools (Wong et al., 2017) and think-aloud sessions (Tahaei et al., 2022b) seek to capture the ephemeral moments in which values are materialized in code. They attempt to reproduce these moments in experimental conditions that allow for causal inference (developers who used a datasheet when analyzing an ethically problematic dataset mentioned ethical concerns earlier and more often, Boyd, 2021); however, these conditions are typically bracketed in time and type of assignment, and the developers are carefully handpicked and compensated for their participation. Thus, to gain access into developers' actual value deliberations, and to do so unobtrusively, some studies focus on naturally-occurring discourse in developers' Q&A forums—specifically, iPhoneDevSDK and XDA (Greene and Shilton, 2018; Shilton and Greene, 2019), StackOverflow (Tahaei et al., 2022a), and /r/androiddev on Reddit (Li et al., 2021). These future-oriented message boards help developers plan apps, solve technical problems, seek feedback, consider events such as new regulations, and share information and opinions about design. In methodological terms, they consist of ready-made discourse unaffected by interviewer or experimental conditions, which is available for scraping and analysis. The discourse analyzed, whether qualitatively or quantitatively, underscores the paradigmatic tension between the "extraction imperative" (Zuboff, 2019) that advises maximal mining and retention of personal information, and the more elusive right to privacy—as well as the power of platforms and regulatory bodies to intervene and set the terms of this conflict. For instance, Greene and Shilton highlight the role of the platform in developers' imaginaries of privacy. Using critical discourse analysis, they find that for iOS developers, privacy was mainly about consent, as required by Apple's App Store; whereas in Android's unregulated free-market open-source ecosystem, privacy was more of an expressive feature that allowed developers to "mark out a role for themselves" (Greene and Shilton, 2018: 1642). Within the Android forum on Reddit, the thematic analysis of Li et al. (2021) finds that exchanges related to personal information were triggered mostly by external requirements to improve privacy, whether by the operating system, the app store, or privacy laws.

In what follows, we propose to get closer to the production of privacy by defining the code-sharing platform GitHub as a site where privacy, as a social value, materializes in code. Specifically, whereas previous research on developers' discourse focused on Q&A forums, whose relationship to the final product is unknown, GitHub README files are directly tied to particular code projects. This immediacy allows us to explore how developers engage with privacy, as a contested concept (Mulligan et al., 2016; Waldman, 2021) and as challenging lines of code (Bednar et al., 2019; Li et al., 2022), in this hands-on environment.

Studying developers' privacy in GitHub README files

The corpus

Discussing hurdles facing social scientists as they attempt to study algorithms, Kitchin (2017) notes that GitHub may provide access to what he and others term the "black box" of code (Bucher, 2016; Pasquale, 2015). Heeding his call, we define the open-source code-sharing platform GitHub as a discursive and practical meeting place of developers' global community of practice—and a site where their privacy ideas and actual code can be studied. In contrast to observations, interviews, assignments, and Q&A forums, GitHub interconnects human and computer languages, juxtaposes code and explanatory text, and allows for an open and unobtrusive analysis of how the former translates into the latter, and back.

GitHub went online in 2008 and has been owned by Microsoft since 2018. According to recent data, it is hosting more than 100 million[1] users and 420 million code repositories (284 million of which are public)—the most widely used platform for open-source software development. GitHub supplements its structurally collaborative ecosystem with social features such as issue tracking, developer following, and repository watching, and various channels of communication like the README file that is attached to most code projects and is used to explain the project, highlight its features, and essentially promote it (Prana et al., 2019).

README files play an important role in GitHub, providing a starting point for platform members to understand and learn about a repository (Wang et al., 2023). These files are the first thing developers see when they explore a project, so developers use them to highlight important features and explain why certain functionalities were adopted. In contrast to comments, which are occasionally made in relation to specific aspects of the code—particular lines, unresolved issues, additions needed—that are typically technical and often limited to 50 characters, README files are extensive documents that may serve as good indicators of their author's professional views and familiarity with programming concepts, languages, and toolkits (Hassan and Wang, 2017).[2] GitHub's official documentation recommends that README files contain the following information: what the project does; why the project is useful; how users can get started with the project; where users can get help with the project; and who maintains and contributes to the project. Thus, as a first step in studying how developers translate their privacy values into GitHub public code projects, we chose to focus on the README files they append to the code they share and in which they describe, among other things, how they implement privacy protections.

Our corpus is based on GitHub code projects created in 2008–2020 by developers from different industries, overcoming time, language, and distance boundaries. Following Shilton and Greene's (2019) focus on texts that contain the word "privacy," we used advanced search options in GitHub's application programming interface (REST API), implemented in dedicated Python scripts, and collected 59,898 README files that contain the word "privacy." To focus the analysis, we then used dedicated Python scripts to extract from the README files the specific paragraphs that contained the word "privacy" and saved them in a text file. We used this text file for the automated analysis and returned to the original README files for the qualitative analysis, as explained below.

Hybrid analytic approach

We adopted a hybrid form of grounded theory research method, combining computational and qualitative analysis, in an attempt to gain analytic insight from the scale of our dataset on one hand, and the depth that the hermeneutic tradition provides on the other. In this effort, we are inspired by both Nelson's (2020) roadmap for conducting computational grounded theory, and interpretive approaches such as critical discourse analysis (Van Leeuwen, 2008) and rhetorical social psychology (Billig, 1996; Potter and Wetherell, 1987) that study how texts construct social meaning and, specifically, how they account for and justify social practices in conflictual situations. Grounded theory (Glaser and Strauss, 1999) is an inductive analytic method based on the identification of categories and themes in the research materials with the aim of reaching a data-informed theoretical understanding of social and cultural worlds. As a labor-intensive method that relies on the expertise of the researcher, however, traditional grounded theory is unsuitable for handling the large datasets that are available now, whose analysis may shed important light on contemporary culture. Thus, in order to cope with our large text-based corpus, we adopted a hybrid form of grounded theory, combining an automated analysis of the entire corpus in the first stage and a close reading of selected texts in the second.

Discussions about multi-method or triangulated approaches in computational social science tend to reduce the qualitative component to manual quality assurance. Considering media content analysis, Zamith and Lewis (2015) aim to develop "better technical systems to conduct the kind of content analysis that truly blends the best of both worlds, human and machine alike" (p. 316). From this perspective, however, the more sensitive machines become, the less human interpretation is required. Shahin (2016) outlines a different methodological balance. His approach uses natural language processing algorithms to extract purposive samples from textual corpora, and an analysis of the extracted texts that draws on discourse analytic methods. Shahin explains that his approach differs from conventional big data research in that it does not depend on modeling techniques for finding patterns in texts but rather, on scholars' theoretical knowledge of a subject and the ways it informs a reflexive interpretation of the findings. Nonetheless, the computational method allows for analyzing datasets that otherwise would be hard, if not impossible, to study qualitatively.

In what follows we adopt a balanced hybrid approach to study the materialization of privacy. Defining GitHub README files as a corpus in which developers, as a community of practice, translate into human language how privacy is written in their code projects, we analyze it both quantitatively and qualitatively.

Quantitative analysis. We extracted all 59,898 paragraphs with the word "privacy" in GitHub README files created in 2008–2020 and analyzed a text file of these paragraphs using an automated process that inductively points to significant themes prevalent in the text without pre-defining any categories (Gunther and Quandt, 2016). Specifically, we applied VOSviewer bibliometric software, which implements Natural Language Processing and Visualization Of Similarities (VOS) algorithms to create a co-occurrence map of the words it contains (Lulewicz-Sas, 2017; Van Eck and Waltman, 2014).[3] The occurrence of noun phrases in a dataset indicates their significance and prominence in a

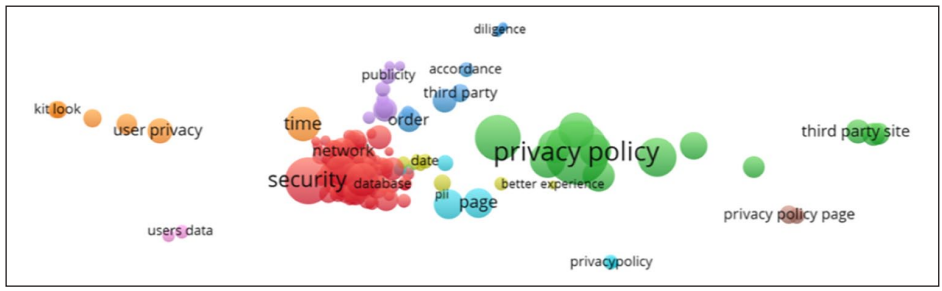

Figure 1. Clusters of privacy themes in the GitHub README files corpus.
Note: A circle represents a word or a term, the circle's size indicates the word's frequency in the text, its color denotes the cluster to which the word belongs, and the distance between the circles signifies the words' lexical relevance.

certain topic area. The co-occurrence of phrases indicates their relationships, the extent of their relevance to each other, and the shape of their interconnections within the dataset. Thus, VOSviewer's map allows us to identify which words are situated near, or co-occur with, the word "privacy" and where they are located in its lexical network.

Analyzing our corpus using VOSviewer produced nine clusters, two large ones and seven smaller, each made up of words that co-occur with the word privacy. We focus our analysis on the two dominant clusters, which represent the main themes in developers' README files privacy discourse. As depicted in Figure 1, the *green* cluster includes 13 words and word pairs, the most common of which is "privacy policy," and the others relate to the control of information ("external site," "third-party site"); the *red* cluster includes 68 words and word pairs, the most frequent is "security," alongside other technology-oriented references that describe technical measures taken to protect users' information (e.g. "algorithm," "anonymity," "differential privacy," "encryption," "protocol," and "transaction"). Table 1 lists the words in each of the two main clusters.[4]

Table 1 suggests that the green cluster contains texts explaining how developers inform users about what they are doing with the data collected. The use of the term "privacy policy" alongside terms such as "third party" and "responsibility" suggests that these README files focus on compliance with rules and regulations and ways to inform users about the handling of their information. This vocabulary relates to a privacy-by-policy approach and Chapter 3 of the General Data Protection Regulation (GDPR) guidelines.[5] It requires that users "should be given the opportunity to make choices about secondary uses of their personal information—those uses that go beyond the original purpose for which the data was provided" (Spiekermann and Cranor, 2009: 77). This approach advises developers wishing to collect personal information for their own or third parties' purposes to explain what information they are collecting and how it may be utilized, and gain users' informed consent. The relatively limited number of circles (words) in this cluster, complemented by their relatively large size (word count), indicate repetition of the same messages and a focus on the topic of (non-) data transfer to a third party.

Table 1. The words and word pairs in the two main clusters.

Cluster	Words in cluster
Green	content, control, external site, link, practice, **privacy policy**, responsibility, service, site, third-party link, third-party site, third-party service provider, website
Red	algorithm, anonymity, approach, attack, authentication, bot, camera, care, challenge, class, client, communication, cloud, coin, communication, community, concept, concern, conversation, course, cryptocurrency, customer, data privacy, database, default, developer, differential privacy, encryption, Etherum, everyone, everything, Firefox, focus, freedom, friend, future, GDPR, GitHub, goal, government, group, identity, implementation, input, instance, IP address, key, log, machine, model, module, network, node, output, patient, privacy concern, privacy, protection, privacy risk, privacy setting, process, program, protocol, **security**, technique, technology, transaction

Note: The words and word pairs are listed in alphabetical order. In bold, the most frequent word/word pair in each cluster.

The red cluster, by contrast, is more technical and deals with measures taken to protect users' information such as anonymization, authentication, differential privacy, and encryption. This vocabulary relates to the privacy-by-design approach (Cavoukian, 2009) and Article 25 of the GDPR guidelines.[6] This approach recommends that developers apply privacy and data protection principles and technical measures from the early stages of software development, to assure "higher levels of privacy to users more reliably and without the need for them to analyze or negotiate privacy policies" (Spiekermann and Cranor, 2009: 79). Echoing findings from interview studies (e.g. Hadar et al., 2018), the size of the security circle and its rich practical vocabulary suggest that developers prioritize security measures when seeking to protect users' privacy. The co-occurrence of this vocabulary with terms such as freedom, privacy risk, and privacy concern underscores the relationships between social values and code—as well as the primary role of GitHub as a site for the articulation, and the study, of these relationships.

Qualitative analysis. We used the automated cluster analysis results as a basis for sampling README files for the qualitative analysis. However, whereas the automated analysis was performed on a corpus that singled out the word "privacy," we now sought to expand our analysis to other privacy-related terms (Li et al., 2021) so as to encompass not only texts that use this particular word, but ones that contain words listed in Table 1, including those that appear in paragraphs that do not mention the word "privacy." We employed Voyant-tools.org's Key-Word-In-Context (KWIC) interface,[7] which displays sentences and paragraphs in a table with each row containing text that appears before and after the analyzed words. The interface allows choosing the number of words before and after the target word, normally 5–50 but up to 500 for a more thorough reading. Using this interface, we reviewed thousands of sentences containing the words from the two main clusters, building a purposive corpus by focusing on paragraphs that present an argument, position statement, conflict or contradiction, or deliberation between alternative options.

Studying this corpus, we find that some README references express a strong commitment to safeguarding users' privacy, ranging from personal "obsession" to whatever is "reasonable and feasible." These developers translate their concern for users' privacy into code by using strong protection measures ("For those who know me, you know I am obsessed with privacy. I believe any application development should adopt a good privacy-minded framework from the start," "We have used industry-standard security measures to protect your personal information from unauthorized access, public disclosure, use, modification, damage or loss. We'll take every reasonable and feasible measure to protect personal information, for example, SSL encryption protects your browser from exchanging data . . ."). [8] In contrast to previous findings that suggest that developers tend to dismiss users' worries, we find README references that acknowledge them and disclose an intent to provide solutions (". . . cater to those who are concerned about how their Personal Identifiable Information [PII] is being used online."). Developers disclose awareness of challenges facing them in protecting users' privacy (". . . higher advertising return in investment often come with negative trade-offs for consumer privacy or security risks of leaking personal data"). This may be presented as a technical inevitability ("Whenever private data is sent to another server on the internet, once the packet is sent over the jungle of the internet, there is no control over who lays their hand on it. This is its nature."). However, other developers contend that it is their duty to protect the users' privacy ("It is your responsibility to respect user's privacy," "Contrary to common practice, privacy policies are not intended for service providers—it is for the service users. This means that the privacy policy should be written with *them* on mind.").

Other README references focus on the obligation to comply with privacy regulations. Here, users' privacy is tied not to the developers' design of protective architecture or challenges therein, but rather, to a reliance on legal defenses ("It's crucial that users understand how their information will be handled, especially in light of the recent updates to legislation including the latest European Union (EU) Regulations," "The legal requirement for businesses is to make available a clear and updated privacy policy explaining how personal information will be used."). Such README references may detail how personal information is used or how to overcome the legal requirements—whether because they dismiss privacy as a concept or a practice, or by using technological tools that generate automatically privacy policy texts and interfaces that seem to meet the required standards ("The plugin automatically adds legal and privacy policy pages to your website," "In this feature, we provide a guided process for creating such policy for your website in accordance with the EU Data Protection Regulation 2018."). References to privacy in the context of compliance are translated into the use of ready-made, plug-and-play privacy policy plugins in the code.

Still referring to compliance, some developers tinker with the regulatory hurdle in their quest for users' private information. Assuming that users do not want to be blocked by repeated messages of consent and are likely to press any button to remove them, their code could gather personal information while appearing to comply ('Track every move your users make. Just make sure to tell them you're doing this in your privacy policy," "The default is to show the Accept button and Privacy Policy button and not the Decline button. In this way we get assumed consent and users cannot opt-out of cookies."). Ironically, it is app platforms that eventually compel developers to align with

their privacy requirements and the local privacy regulations lest they be prevented from integrating with them. Under these circumstances, developers comply with platforms' demands not because they deem privacy important, but to obtain their desired functionality ("I need privacy policy feature for Facebook because I am toying with their API," "Google told me that my app must have a privacy policy otherwise it would be removed from the app store, so I added it.").

However, a broader reading of the texts complicates this binary distinction, suggesting that even developers who profess staunch commitment to privacy operate in an ecosystem that constantly threatens it. Such dilemmatic accounts (Billig, 1996) may be found as we move from paragraphs to files, where developers outline their vision, often as a resolution of tension between conflicting demands; for instance, as part of a "journey toward reinventing privacy from the ground up," a README file opens by saying:

> We believe every human on the planet should have free access to the world's information and content. This is powered by a three way value exchange between a media owner publishing free content, a consumer consuming it and an advertiser paying for the chance to connect with the media owner's audience . . . We envision a world . . . where consumer privacy, security, and governance are incorporated into the fabric of the code base while enabling the necessary business use-cases to effectively keep the world's information and content free for everyone.

The topic of this preamble appears to be "every human on the planet" but in fact it is concerned with the operation of an economic system that relies on these humans' data as currency. Employing a discourse of transformation allows the authors to blur the fact that their code does not offer a way out or role reversal: In the envisioned utopia, users' personal information remains the currency and the active players remain the media owner and the advertiser. The authors do mention later in the file that "privacy-preserving measurement is a fast-evolving field," but they fail to explain what solutions they seek from it, or how their current project handles the breeches that they critique in the system.

On a more mundane level, this file describes a feature that allows for anonymous confessions in a social media platform: "Your posts are anonymous. No one will know who you are. However, to prevent abuse of this feature, certain moderators may be able to access the posts' user info . . ." recognizing potential misuses, the file immediately cautions the moderators: "Do not use this feature to stalk your members or for your own benefits . . . by following the steps below, you agree to use the feature with good faith." Thus the promise of an anonymous confession relies on trust in the moderators' cooperation to ensure it. Another developer declares:

> Security is one aspect of privacy protection, it doesn't complete the puzzle. Applications, whether web-based, mobile, or otherwise, are often developed to gather telemetry, statistics, usage, and personally identifiable information such as names and email from sign-up forms, IP addresses, browser information, and tracking cookies. As for the latter, many websites use Google analytics, which also adopts tracking cookies through embedded JavaScript and cookies.

This well-intentioned statement prefaces a feature that would enable interactivity in static websites by embedding a Twitter tweet for comments: "Simply add the Tweet ID, which is part of the Twitter URL for a tweet . . . Social media icons can be added easily, for all popular social media platforms, displaying the icons in the sidebar." As Gerlitz and Helmond (2013) show, however, such icons allow the translation of user engagement anywhere on the web into numbers on button counters, which the platforms can both trade and use as tracking devices. Rejecting commentary plugins to protect users' privacy, then, this developer trusts it with Twitter and its less visible partners (van der Vlist and Helmond, 2021).

Concluding comments

Toward a materialist understanding of information privacy, this article makes three contributions: First, it critically maps the research on developers' privacy, outlining a hybrid method that resolves some of the difficulties we have identified. Second, it develops a rationale for considering GitHub as a site for the study of the production of privacy, highlighting its key position in developers' practice and the immediate link it provides between social values and digital products. Finally, the article analyzes developers' privacy discourse, distinguishing between design and consent approaches and noting the weakness even of the former in protecting users' privacy. Recent scholarship has suggested developers are key for understanding how social values on one hand, and work practices on the other, are translated into computer software. The programs, apps, and interfaces they produce not only afford specific uses and preclude others but—through users' actual engagement with the screen—they naturalize the worlds and worldviews within which they were written. In this way, even though this scholarship typically originates in and seeks to improve engineering and ethics education, it implies that developers' values and work practices gain a material form. By contrast, this article adopts digital materialism as a theoretical starting point, arguing that privacy needs to be studied as a concept, as software and as hardware; and that GitHub—as a platform in which developers write, share, and discuss privacy code—may offer important insights for a study that assumes a socio-material perspective.

As we have suggested, most scholarship in this field (with some noted exceptions) portrays developers as practitioners torn between strong economic incentives for maximal information mining, and a tentative valuation of minimal extraction and retention; torn between ready-made solutions under project time constraints, and an interest in alternative designs and architectures. Our analysis reinforces this picture, identifying two overarching approaches to privacy—one committed, viewing it as a value to uphold, and the other regarding it as an impediment to circumvent, elaborating on Spiekermann and Cranor's (2009) foundational distinction in "engineering privacy" between privacy by policy and privacy by architecture. In addition, however, the analysis suggests that even commitment to privacy does not necessarily translate into effective privacy protection.

Thus, our findings engage with previous scholarship in several ways. First, in interviews and assignments, developers are repeatedly portrayed as inadequate, passive

objects of pressures—lacking knowledge, time, and depth. By contrast, the README files authored by developers reveal a sense of agency: developers appear not as hurried workers merely following demands and schedules, but as creative contributors to the production process. Not unlike the veteran, opinionated developers in Greene and Shilton's study (Greene and Shilton, 2018; Shilton and Greene, 2019), these developers articulate their socio-technical philosophy, present a vision and express their reservations about how personal information is typically handled and how their code is designed to improve it. Second, studies have observed that when asked about privacy, developers typically respond in terms of security (Hadar et al., 2018). Our analysis shows that developers raise the issue of information privacy even when not prompted, and that they discuss privacy not only in terms of data protection, but also have much to say about it both technically and normatively, even if their code may inadvertently violate it. Third, previous studies have distinguished privacy champions (Tahaei et al., 2021) and legal experts (Horstmann et al., 2024) from ordinary developers, critiquing the divide between them as well as the latter's indifference to privacy concerns. Our analysis questions these distinctions, suggesting both that advocating for privacy may not be limited to organized teams or legal departments—and that privacy breaches may occur even in code written by privacy enthusiasts.

Finally, and crucially, whereas previous studies have considered developers separate from the actual code they wrote, our focus on GitHub README files allowed us to examine developers' notions of privacy as related to particular pieces of code they created. Thus we find that in this context—an ecosystem that interconnects social values to computer language—privacy is taken into account, whether as a goal in itself or as a consideration on the way to achieve another goal—and whether or not user's privacy is in fact secured. Indeed, in light of the translation immediacy GitHub offers—as well as the abundance of public repositories on this platform and the analytic method we developed—GitHub opens new directions for studying developers' privacy. Historicizing developers' views and practices is an important endeavor, examining how their code—and their accounts—have evolved over time. It would also be worthwhile to explore developer's privacy perceptions in relation to emerging technologies such as Artificial Intelligence (AI). Focusing on developers working in big-tech companies may reveal how they express themselves in open-source software. Finally, in follow-up studies, we explore how developers refer to privacy in their GitHub profiles; and how code authored by hackers incorporates privacy protections.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Israel Science Foundation under grant number 651/20.

ORCID iD

Rivka Ribak https://orcid.org/0000-0003-0548-7885

Notes

1. https://octoverse.github.com/
2. In May 2020, GitHub introduced a beta version of a discussion feature where community members could "ask and answer questions, share updates, have open-ended conversations, and follow along on decisions affecting the community's way of working" (https://docs.github.com/en/discussions). This took place during the final stages of data collection for the current research, but it may offer an intriguing forum-type research field in the future.
3. https://www.vosviewer.com/
4. The files used to create the map are accessible on GitHub under the username githubprivacyresearch.
5. https://gdpr-info.eu/chapter-3/
6. https://gdpr-info.eu/art-25-gdpr/
7. https://voyant-tools.org/docs/#!/guide/about
8. The excerpts were slightly re-worded to protect developers' anonymity.

References

Acar Y, Fahl S and Mazurek ML (2016). You are not your developer, either: a research agenda for usable security and privacy research beyond end users. In: *2016 IEEE cybersecurity development (Secdev)*, Boston, MA, 3–4 November, pp. 3–8. New York: IEEE.

Alomar N and Egelman S (2022) Developers say the darnedest things: privacy compliance processes followed by developers of child-directed apps. *Proceedings on Privacy Enhancing Technologies* 4: 24.

Balebako R, Marsh A, Lin J, et al. (2014) The privacy and security behaviors of smartphone app developers. In: *Proceedings workshop on usable security (USEC'14)*, San Diego, CA, 23 February. Reston, VI: The Internet Society.

Bednar K, Spiekermann S and Langheinrich M (2019) Engineering privacy by design: are engineers ready to live up to the challenge? *The Information Society* 35(3): 122–142.

Billig M (1996) *Arguing and Thinking: A Rhetorical Approach to Social Psychology*. Cambridge: Cambridge University Press.

Boyd KL (2021) Datasheets for datasets help ML engineers notice and understand ethical issues in training data. *Proceedings of the ACM on Human-computer Interaction* 5(CSCW2): 1–27.

Boyd KL and Shilton K (2021) Adapting ethical sensitivity as a construct to study technology design teams. *Proceedings of the ACM on Human-computer Interaction* 5(GROUP): 1–29.

Bucher T (2016) Neither black nor box: ways of knowing algorithms. In: Kubitschko S and Kaun A (eds) *Innovative Methods in Media and Communication Research*. London: Palgrave Macmillan, pp. 81–98.

Canedo ED, Bandeira IN, Calazans ATS, et al. (2023) Privacy requirements elicitation: a systematic literature review and perception analysis of IT practitioners. *Requirements Engineering* 28(2): 177–194.

Casemajor N (2015) Digital materialisms: frameworks for digital media studies. *Westminister Papers in Communication and Culture* 10(1): 4–17.

Cavoukian A (2009) Privacy by design: the 7 foundational principles. *Information and Privacy Commissioner of Ontario, Canada* 5: 12.

Coleman EG (2010) Ethnographic approaches to digital media. *Annual Review of Anthropology* 39: 487–505.

Frenkel M (2005) The politics of translation: how state-level political relations affect the cross-national travel of management ideas. *Organization* 12(2): 275–301.

Friedman B (1996) Value-sensitive design. *Interactions* 3(6): 16–23.

Geertz C (2005) Deep play: notes on the Balinese cockfight. *Daedalus* 134(4): 56–86.
Gerlitz C and Helmond A (2013) The like economy: social buttons and the data-intensive web. *New Media & Society* 15(8): 1348–1365.
Glaser BG and Strauss AL (1999) *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Chicago, IL: Aldine.
Gottlieb B and Cornet M (2021) Digital materialism and cyber-automation. In: Wahal E (ed.) *Unboxing AI: Understanding Artificial Intelligence*. Milan: Fondazione Giangiacomo Feltrine, pp. 33–40.
Greene D and Shilton K (2018) Platform privacies: governance, collaboration, and the different meanings of "privacy" in iOS and Android development. *new media & society* 20(4): 1640–1657.
Gunther E and Quandt T (2016) Word counts and topic models: an overview of automated text analysis methods for digital journalism research. *Digital Journalism* 4(1): 75–88.
Hadar I, Hasson T, Ayalon O, et al. (2018) Privacy by designers: software developers' privacy mindset. *Empirical Software Engineering* 23(1): 259–289.
Hassan F and Wang X (2017) Mining readme files to support automatic building of java projects in software repositories. In: *2017 IEEE/ACM 39th international conference on software engineering companion (ICSE-C)*, Buenos Aires, 20–28 May, pp. 277–279. New York: IEEE.
Hiller M (2015) Signs of "the Times": the software of philosophy and the philology of software. *Digital Culture & Society* 1(1): 151–163.
Horstmann SA, Domiks S, Gutfleisch M, et al. (2024) "Those things are written by lawyers, and programmers are reading that": mapping the communication gap between software developers and privacy experts. *Proceedings on Privacy Enhancing Technologies* 24(1): 151–170.
Jørgensen RF (2018) Framing human rights: exploring storytelling within internet companies. *Information, Communication & Society* 21(3): 340–355.
Kitchin R (2017) Thinking critically about researching algorithms. *Information, Communication & Society* 20(1): 14–29.
Kotliar DM (2020) The return of the social: algorithmic identity in an age of symbolic demise. *New Media & Society* 22(7): 1152–1167.
Li T, Louie E, Dabbish L, et al. (2021) How developers talk about personal data and what it means for user privacy: a case study of a developer forum on reddit. *Proceedings of the ACM on Human-Computer Interaction* 4(CSCW3): 1–28.
Li T, Reiman K, Agarwal Y, et al. (2022) Understanding challenges for developers to create accurate privacy nutrition labels. In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, New Orleans, LA, April 29–May 5, pp. 1–24. New York: ACM.
Lulewicz-Sas A (2017) Corporate social responsibility in the light of management science—bibliometric analysis. *Procedia Engineering* 182: 412–417.
Mulligan DK, Koopman C and Doty N (2016) Privacy is an essentially contested concept: a multidimensional analytic for mapping privacy. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374(2083): 20160118.
Nelson LK (2020) Computational grounded theory: a methodological framework. *Sociological Methods & Research* 49(1): 3–42.
Nissenbaum H (2005) Values in technical design. In: Mitcham C (ed.) *Encyclopedia of Science, Technology, and Ethics.* New York: Macmillan, pp. 66–70.
Parikka J (2015) *A Geology of Media*. Minneapolis, MN: University of Minnesota Press.
Pasquale F (2015) *The Black Box Society: The Secret Algorithms That Control Money and Information*. Cambridge, MA: Harvard University Press.
Passoth JH (2019) From hardware to software to runtime: the politics of (at least) three digital materialities. In: Kissmann U and van Loon J (eds) *Discussing New Materialism: Methodological Implications for the Study of Materialities*. New York: Springer, pp. 173–189.

Peixoto M, Ferreira D, Cavalcanti M, et al. (2020) On understanding how developers perceive and interpret privacy requirements research preview. In: *Requirements engineering: foundation for software quality:26th international working conference REFSQ 2020*, Pisa, 24–27 March, pp. 116–123. Berlin: Springer International Publishing.

Peters JD (2017) "You mean my whole fallacy is wrong": on technological determinism. *Representations* 140: 10–26.

Potter J and Wetherell M (1987) *Discourse and Social Psychology: Beyond Attitudes and Behaviour*. Thousand Oaks, CA: Sage.

Poulsen SV and Kvåle G (2018) Studying social media as semiotic technology: a social semiotic multimodal framework. *Social Semiotics* 28(5): 700–717.

Prana GAA, Treude C, Thung F, et al. (2019) Categorizing the content of GitHub README files. *Empirical Software Engineering* 24: 1296–1327.

Reichert R and Richterich A (2015) Introduction. Digital Materialism. *Digital Culture & Society* 1(1): 5–17.

Ribak, R. (2019). Translating privacy: developer cultures in the global world of practice. *Information, Communication & Society*, 22(6): 838–853.

Senarath A and Arachchilage NAG (2018) Understanding software developers' approach towards implementing data minimization. In: *USENIX symposium on usable privacy and security (SOUPS)*, Baltimore, MD, 12–14 August.

Shahin S (2016) When scale meets depth: integrating natural language processing and textual analysis for studying digital corpora. *Communication Methods and Measures* 10(1): 28–50.

Shilton K (2013) Values levers: building ethics into design. *Science, Technology & Human Values* 38(3): 374–397.

Shilton K and Greene D (2019) Linking platforms, practices, and developer ethics: levers for privacy discourse in mobile application development. *Journal of Business Ethics* 155(1): 131–146.

Spiekermann S and Cranor LF (2009) Engineering privacy. *IEEE Transactions on Software Engineering* 35(1): 67–82.

Spiekermann S, Korunovska J and Langheinrich M (2018) Inside the organization: why privacy and security engineering is a challenge for engineers. *Proceedings of the IEEE* 107(3): 600–615.

Tahaei M, Frik A and Vaniea K (2021, May) Privacy champions in software teams: Understanding their motivations, strategies, and challenges. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (pp. 1–15).

Tahaei M, Li T and Vaniea K (2022a) Understanding privacy-related advice on stack overflow. *Proceedings on Privacy Enhancing Technologies* 2022(2): 114–131.

Tahaei M, Ramokapane KM, Li T, et al. (2022b) Charting app developers' journey through privacy regulation features in ad networks. *Proceedings on Privacy Enhancing Technologies* 2022(3): 33–56.

Takhteyev Y (2012) *Coding Places: Software Practice in a South American City*. Cambridge, MA: MIT Press.

Van der Linden D, Anthonysamy P, Nuseibeh B, et al. (2020) Schrödinger's security: opening the box on app developers' security rationale. In: *Proceedings of the ACM/IEEE 42nd international conference on software engineering*, Seoul, 27 June–19 July, pp. 149–160. New York: ACM.

Van der Vlist FN and Helmond A (2021) How partners mediate platform power: mapping business and data partnerships in the social media ecosystem. *Big Data & Society* 8(1): 20539517211025061.

Van Eck NJ and Waltman L (2014) CitNetExplorer: a new software tool for analyzing and visualizing citation networks. *Journal of Informetrics* 8(4): 802–823.
Van Leeuwen T (2008) *Discourse and Practice: New Tools for Critical Discourse Analysis*. Oxford; New York: Oxford University Press.
Wajcman J (2019) How silicon valley sets time. *New Media & Society* 21(6): 1272–1289.
Waldman AE (2021) *Industry Unbound: The inside Story of Privacy, Data, and Corporate Power*. Cambridge: Cambridge University Press.
Wang T, Wang S and Chen THP (2023) Study the correlation between the readme file of GitHub projects and their popularity. *Journal of Systems and Software* 205: 111806.
Wong RY, Mulligan DK, Van Wyk E, et al. (2017) Eliciting values reflections by engaging privacy futures using design workbooks. *Proceedings of the ACM on Human-Computer Interaction* 1(CSCW): 1–26.
Zamith R and Lewis SC (2015) Content analysis and the algorithmic coder: what computational social science means for traditional modes of media analysis. *The ANNALS of the American Academy of Political and Social Science* 659(1): 307–318.
Zhao S, Djonov E and Van Leeuwen T (2014) Semiotic technology and practice: a multimodal social semiotic approach to PowerPoint. *Text & Talk* 34(3): 349–375.
Zuboff S (2019) Surveillance capitalism and the challenge of collective action. *New Labor Forum* 28(1): 10–29.

Author biographies

Keren Levi-Eshkol is a doctoral student in the Department of Communication at the University of Haifa. Her study focuses on the translation of developers' values into software. She explores how developers' ideas about information privacy are shaped in GitHub repositories by employing advanced natural language processing (NLP) tools.

Rivka Ribak is an Associate Professor in the Department of Communication at the University of Haifa. She is interested in media use and non-use in different cultural contexts and in emergent practices of digitization and photography. Her most recent project explores how developers' ideas about information privacy are shaped in cross-cultural encounters, and how such encounters are translated into code.