Platform privacies: Governance, collaboration, and the different meanings of “privacy” in iOS and Android development

Daniel Greene
Microsoft Corporation, USA

Katie Shilton
University of Maryland, USA

Abstract
Mobile application design can have a tremendous impact on consumer privacy. But how do mobile developers learn what constitutes privacy? We analyze discussions about privacy on two major developer forums: one for iOS and one for Android. We find that the different platforms produce markedly different definitions of privacy. For iOS developers, Apple is a gatekeeper, controlling market access. The meaning of “privacy” shifts as developers try to interpret Apple’s policy guidance. For Android developers, Google is one data-collecting adversary among many. Privacy becomes a set of defensive features through which developers respond to a data-driven economy’s unequal distribution of power. By focusing on the development cultures arising from each platform, we highlight the power differentials inherent in “privacy by design” approaches, illustrating the role of platforms not only as intermediaries for privacy-sensitive content but also as regulators who help define what privacy is and how it works.

Keywords
Apple, collaboration, Google, governance, mobile media, platforms, privacy, values in design

Corresponding author:
Daniel Greene, Microsoft Corporation, 1 Memorial Drive, Cambridge, MA 02142, USA.
Email: dgreene@microsoft.com
Introduction

Two senior privacy engineers spoke at Apple’s 2016 Worldwide Developer Conference (WWDC) to developers designing software applications (apps) for the company’s platform. In their talk, “Engineering Privacy for Your Users,” the fourth slide read, “Your users + you + privacy = ☺” (Pease and Freudiger, 2016). There is an important actor missing here: Apple itself. The “you” of this equation, developers, must produce apps that pass Apple’s review process. The WWDC presentation was intended to train developers on how Apple defines privacy and how that value can be built into apps.

Apple is not alone in training app developers, who work outside the company but create products for the platform, in its values. The Google Developers blog, for example, regularly posts new tools and feature sets that model Google’s approach to privacy (e.g. Khazanie and Matias, 2016). Developer values are also governed through more quotidian means. Developers learn what privacy means through their everyday work practices, as they interact with iOS or Android’s technical features, its human representatives, and other developers working through similar products and problems. In turn, developers must make decisions about how best to incorporate platform values into their products to gain the access to markets provided by both platforms. Formal presentations and publications are just the tip of a values governance iceberg.

This article explores the rest of the iceberg or at least the portion devoted to mobile application development. Mobile application developers (“devs”) range from hobbyists to employees of multinational corporations (Stack Overflow, 2016). They develop apps that collect a diverse array of personal data, such as geolocation and call logs. Devs sell these apps through platform stores, such as Google’s Play Store and Apple’s App Store. During this process, they navigate the rules and regulations of these platforms. Since the introduction of the App Store in 2008 and the Android App Market in 2009 (replaced by the Google Play Store in 2012), mobile apps have become a primary means by which consumers engage with digital content. Between 2013 and 2015, mobile app usage grew by 90% and contributed to 77% of the total increase in digital media time spent by US consumers. Time on mobile devices now accounts for 2 out of every 3 minutes spent with digital media, and mobile apps themselves now constitute just over half of those minutes (comScore, 2015).

The growth of the app-based model of software, coupled with the granular personal data that mobile apps can collect, makes app development an important site to explore the concept of privacy. Data collected by apps are largely unregulated in the United States (Rubinstein, 2011) and may be sold to advertisers, shared with strategic partners, or given to analytics companies. How privacy is defined and implemented—or not—in mobile application development has broad implications for consumers of digital media and the future of the internet more generally. The ways that developers understand, debate, and attempt to enact privacy shape app design and use, and determine what data are available to other actors within the mobile application ecosystem. Devs must also weigh the importance of privacy against other values such as efficiency, cost, security, and accessibility. To understand how devs legitimate privacy as a design criterion and define it in practical terms, we conducted a critical discourse analysis of privacy discussions in Android and iOS developer forums. Our guiding questions were the following:
1. How is privacy authorized as a design problem in contrasting development communities?
2. How is privacy defined among mobile application developers?
3. How do mobile platforms, through technical or regulatory means, shape these definitions?

We find that privacy is a frequent topic of conversation in iOS and Android developer forums. But developers encounter, define, and legitimate privacy in fundamentally different ways within these platforms. The findings illustrate how definitions of privacy change as a function of developers' subjectivity (i.e. their sense of self and their values within particular contexts) and how that subjectivity is produced through platform governance. Platform governance consists of a mix of formal restrictions and policy alongside informal training as developers debate the “right” way of doing things. It utilizes both negative and positive power: platform governance discourages or prohibits developers from making specific design decisions and encourages or teaches them to make other ones more in line with platform values. The centralized distribution points of app marketplaces controlled by each platform and the distinct development cultures that arise around each platform are powerful influences.

The style and impacts of governance differ between platforms. In the iOS ecosystem, Apple functions as a regulator and a gatekeeper, defining and legitimating privacy. The meaning of “privacy” shifts as developers try to interpret and test the limits of Apple’s policy guidance. Developers collaborate to determine what Apple means by “privacy,” so that they can bring their app to market within Apple’s gated ecosystem. In Google’s open-source Android ecosystem, privacy is a feature: a way for open source advocates, activists, and tinkerers to respond to the unequal distribution of power in a data-driven economy and mark out a role for themselves distinct from the platform for which they are developing software. Devs design privacy features to build identity and community, but these privacy-enhancing features are largely only accessible to people with high levels of technological literacy and interest.

Privacy is a critical challenge for mobile development (Balebako and Cranor, 2014; Urban et al., 2012), and this article adds to the literature on privacy in mobile media. Finally, the different meanings of “privacy” in Android and iOS ecosystems produce different consequences for users: broadly, iOS encourages accessible security, while Android affords privacy creativity. Understanding these challenges can help us shape better guidelines for privacy by design, inform regulation of platforms, and broach challenges to the adoption of privacy by design principles.

**Background**

Current US approaches to mobile data protection rely on self-regulation through privacy by design: encouraging developers to proactively implement best-practice privacy features to protect sensitive data (Federal Trade Commission (FTC), 2012). Precisely what “privacy” means for design, however, is contested (Mulligan et al., 2016). Technical research has produced a range of methods and tools to give users control over data on their mobile devices (e.g. Cáceres et al., 2009; Jeon et al., 2012). However, privacy
measures are particularly difficult to implement in the mobile technology space, where challenges range from space restrictions that limit privacy notices (Schaub et al., 2015) to the conflict between consumers’ privacy interests and platforms’ advertising-based revenue models (Leontiadis et al., 2012). More broadly, privacy scholars argue that meaningful privacy protection is more complicated than enabling user control over data (Cohen, 2012; Nissenbaum, 2009). Nissenbaum’s influential theory of contextual integrity specifies that privacy expectations depend upon complex contextual and situational norms. Identifying and complying with these norms are challenges for developers.

Research has investigated consumers’ perceptions of mobile privacy and has found wide gaps between consumer expectations and data-sharing realities (Lin et al., 2012; Martin and Shilton, 2016; Sadeh et al., 2009). As predicted by contextual integrity, consumers expect some data uses in constrained contexts; for example, they expect navigation apps, but not games, to collect their location (Martin and Shilton, 2016). Research has also shown that consumers assume platforms test and vet apps when they do not (Kelley et al., 2012). Simultaneously, researchers have quantified the extraordinary amount of information sharing in the mobile economy. For example, Zang et al. (2015) found that iOS and Android apps widely share personally identifiable information (PII), behavior data, and location data with third parties. They also discovered clear differences between platforms in the amount and nature of data shared. Seventy-three percent of Android apps shared PII with third-party domains compared to 16% of iOS apps. While 47% of iOS apps shared location data with third-party domains compared to 33% of Android apps. What this research does not reveal, however, is how and why developers make starkly different decisions about data sharing within each platform.

The hows and whys of developer decision-making matter because privacy by design positions developers and mobile companies as ethical agents. In a loose US regulatory context subject to contested privacy definitions and conflicting values, design decisions become the political and ethical terrain on which privacy materializes (Rubinstein, 2011). And as developers become ethical agents, platforms emerge as de facto regulators. Platforms block, promote, flag, ban, feature, and edit the content they host (Gillespie, 2015). Research literature on platforms has focused either on intermediaries for user-produced content, such as Facebook and YouTube (Gillespie, 2010), or on more materialist approaches to the “full stack” of a technology’s lifecycle, from design through its media representations and user experiences (Bogost and Montfort, 2008). We take a middle road, focusing on the regulation of technical designs rather than on art or speech. We are concerned with the explicit regulation and implicit training Google and Apple provide to shape the design approach app developers—who do not work for these firms—take in producing software for each platform. Platforms create development ecosystems where devs, advertisers, and expert users collaborate to shape distinct visions of what privacy is and how it works. In both platforms, devs must abide by platform rules: in return for access to a centralized portal that provides access to customers and lowers distribution costs, developers must accept more centralized forms of control (Holzer and Ondrus, 2011).

The platforms studied here are Apple’s mobile-operating system, iOS, and the Google equivalent, Android. These are the two most popular mobile platforms in the United States, accounting for 63% of the US mobile phone market (Pew Research Center, 2015).
Android and iOS are not just competitors but also fundamentally different environments for developers and consumers. The iPhone is the only phone running iOS; in contrast, Android is available on a variety of handsets with varying capabilities. The iPhone is a famously closed development environment encouraging purpose-built software and minimal user tinkering in the name of curated and a unified mobile experience. Apple requires an approval process for all applications in its store, and developers must comply with Apple’s content, privacy, and security policies to gain approval (Spencer, 2016). Zittrain (2008) termed this environment a “walled garden” reminiscent of pre-Web internet services like CompuServe and America Online (AOL). In contrast, Android is an open-source platform that handset designers, devs, and hobbyists can modify as they please. Although placing apps in the Play Marketplace does require developers to agree to Google’s Developer Distribution Agreement (Google Play, 2016), which includes basic privacy guidelines, the agreement explicitly states that Google does not “undertake an obligation to monitor the Products or their content.” This open approach to design was crucial to Android’s rapid global uptake, just as a rhetoric of “openness” was of critical importance to Google’s marketing of Android (Goggin, 2012), helping Android grow into a much larger ecosystem (Stack Overflow, 2016).

With this literature on platform governance as our backdrop and the collaborative space of developer forums as our focus, we reveal the everyday interactions between devs and the platform, and describe the ways devs’ privacy design decisions are shaped by the platform in which they work. Platforms exert not just negative power (hiding or blocking content or users) but also positive power: training devs on what privacy means and why it is important.

**Method**

To trace the work of privacy construction in each ecosystem, we conducted a critical discourse analysis of mobile developer forums. This is a qualitative method for analyzing how participants talk about their social practices (Leeuwen, 2008). Discourse analysis looks for the ways that written texts recontextualize social practice by representing actors, action, legitimacy, and purpose. Critical discourse analysis indicates a focus on the power differentials driving the recontextualization of social practice. Here, we are concerned with two distinct fields of power. First, we are interested in devs’ agency to design specific ideas of what privacy is and how it works into mobile apps. Second, we examine platforms’ ability to regulate devs’ definitions of privacy. Platform regulation surfaces through devs’ descriptions of interactions with platforms’ technical features and encounters with platforms’ human representatives. Platform regulation also emerges in the cultural imaginary produced by conversation among devs: their shared vision of what Apple or Google value and where, how, and why platforms exert those values.

We analyzed data from two forums: iPhoneDevSDK and XDA. The iPhoneDevSDK forum is one of the most popular and widely used iOS developer forums. It features topics such as code sharing, programming tutorials, open discussion, and marketing guidance. Unlike other Apple-related forums, it is meant primarily for devs and focuses on development advice. Unlike the official Apple developer forum, it does not require an Apple-issued developer key to participate. This means participants have diverse development
experience and purposes for participating, and that non-devs (e.g. advertising network representatives searching for potential clients) sometimes participate. Devs liked the forum’s independence from Apple. iPhoneDevSDK offers personalization measures (e.g. a profile page with a record of participation and personalized icons), an emoji-based reaction system that lends character to up-votes on posts and a reputation system. Participants earn badges for engagement milestones (e.g. 500 posts and 5-year anniversaries) or for reactions from others (e.g. 25 “Insightful” reactions to a participant’s content grant them an Insightful badge).

XDA is a massive collection of forums on a variety of technology topics and includes within it the largest Android developer forums on the English-language Web. It features many of the same technical topics as iPhoneDevSDK but widens its participant base. Because of XDA’s size and Android’s open operating system, XDA was characterized by a high degree of collaboration between developers and “hobbyists”—power users who could not only critique devs’ code but also often “rooted” their phones to customize the operating system or download software from alternative markets outside of the Play Store (including some apps posted directly within XDA). Compared with iPhoneDevSDK, XDA was more diverse in terms of professional background and geographic location, drawing global participants with all levels of expertise and interest. The forum also supported a detailed reputation system, featuring membership ranks based on activity.

We initially approached these contrasting forums because previous, unpublished pilot research indicated that developers frequently specialize in a single platform, forming distinct communities of practice. A survey of mobile developers conducted by the site Stack Overflow (2016) validates this insight. However, a limitation of our method in this research is that we are unable to track (generally pseudonymous) developers’ work histories. Some developers in the forums discussed switching platforms, and others may work across both platforms.

In each forum, we collected data based on the phenomenon of interest. We searched for threads which contained the term “privacy” and analyzed those that included a discussion of privacy as it applied to application development. We discarded threads where privacy was used as a keyword in an advertisement for an app or job ads that promised privacy for job applicants. On iPhoneDevSDK, we found 155 results in June 2015 (ranging from 2009 to 2015) that fit these criteria. XDA is a much larger community, reflecting the larger numbers of Android developers (Stack Overflow, 2016). To narrow our search and ensure results contained active discussions, we limited our “privacy” search to threads containing at least two replies within XDA’s App Developers, Android Wear, or Android Development and Hacking forums. The search was performed in October 2015 and yielded 485 results (ranging from 2009 to 2015). To balance our analysis with the smaller iPhoneDevSDK, we sampled every third result. We exported sampled threads from both forums to Dedoose for coding.

Both authors performed a first read of the data to generate initial thematic codes focused on privacy definitions. Right away we noticed legal definitions, platform-specific definitions, and ethical definitions of privacy. Reading XDA and iPhoneDevSDK threads also revealed the distinct affordances of each site, the conversational styles in each forum, and distinctions in the professional concerns displayed in each. The reading
process also surfaced the powerful role of the Android and iOS platforms. This drove our generation of codes that analyzed privacy discussions not just as values-driven debates but as functions of the development cultures of specific platforms. Once initial codes were established, we divided the data set in half and coded threads separately, reviewing each other’s codes in weekly meetings to ensure mutual understanding and thematic coherence. During this process, the code set grew to include emphasis on the “opposite” of privacy (generally data collection and personalization needs), ways that privacy was authorized and legitimated, and conceptions of other actors in the ecosystem (e.g., Google, representatives of firms producing popular software development kits [SDKs], and users).

Our university’s Institutional Review Board (IRB) certified that forum data qualified as public data and, thus, did not require further IRB review. However, we believe that quoting participants violates the contextual integrity of the forum space; participants may not expect their posts to be used for research. To minimize this violation, we have altered participant handles and slightly altered quotations within this article to reduce the search ability of specific exchanges. Alterations preserve the original meaning of posts, and all analyses were conducted on the original, unaltered quotations. We have also announced our ongoing work with posts on each forum and offered a survey to participants as part of follow-up research.

Findings

Privacy discussions occurred in both ecosystems but were more complex and engaged in the iOS ecosystem. In each ecosystem, devs defined privacy in ways that responded to regulatory moves made by the platforms, the affordances of the marketplace models employed by each platform, and the distinctive culture of each ecosystem. In what follows, we first show how devs authorized privacy, invoking third parties whose expertise, moral standing, or regulatory power impressed upon devs the need to design for privacy. Once privacy became a legitimate concern, it had to be operationalized in concrete policy and design decisions. We then examined how devs in each ecosystem produced distinct definitions of what privacy is and how it works.

Authorizing privacy

Developers viewed a number of actors as authorities who could sanction privacy violations, block access to markets, or even exercise moral claims. These authorities helped to make privacy a legitimate design concern in both ecosystems. Discussion about these authorities—the federal government, the app platform and app users—were broadly similar across iOS and Android, with the important exception of the nature of the authority lent to Google and Apple.

Government authority. References to government authority were rare and often non-specific. Most references to government authority invoked privacy as an overarching political value to which all US devs must adhere, as in iOS dev Klicker’s 2009 declaration that “The founding fathers ensured we had rights to personal privacy …”
Occasionally, guidance was more specific. As iOS dev bossapp wrote in 2013, referring to a quote from the Federal Trade Commission (FTC) in a popular press article,

The FTC quote says it all: if you’re developing mobile apps, you have to give the straight story about what your app can do and be transparent about your privacy practices.

The FTC was occasionally invoked in Android forums as well. In a thread which began in November 2012 and updated continuously through May 2015, entitled “[GUIDE] Some incredibly simple things to protect YOUR PRIVACY!,” CornWall wrote in 2013,

The FTC (Federal Trade Commission is the chief American privacy regulator) issued a report (Feb 1, 2013) that show how to boost mobile privacy. Find it at: http://www.ftc.gov/opa/2013/02/mobileprivacy.shtm

Government authorization of privacy could work in the other direction as well; sometimes the government (almost always imagined as the US federal government) was positioned as a party to protect against. XDA developers, in particular, authorized privacy by invoking government surveillance. Edward Snowden’s National Security Agency (NSA) leaks were a popular topic of conversation, and devs often discussed building privacy measures to protect themselves. For example, in response to Cornwall’s invocation of the FTC to authorize privacy, NY Limited said, “Those jerks just wanna quietly scoop even more data!”

User authority. In both iOS and Android ecosystems, devs cited users to authorize privacy as a legitimate design concern. User authorization was sometimes framed as instrumental: users might reject privacy-invasive apps. In other cases, user authorization was framed as dutiful: many devs felt a charge to care for or protect users.

Protective measures stemming from an ethic of care for users were common. This was most often described as an imperative to notify users of data collection measures and garner their consent. But higher standards of care also appeared, particularly in the scrutiny of unfamiliar pieces of software. For example, when iOS dev uNbOuNd considered installing third-party advertising in 2010, they wrote,

Hi, I was about to put [third party ad software] in an app I’m finishing, but I have a serious concern. AddressBook framework is required for having [company] ads. Excuse me?!!! I’m very concerned regarding my user’s privacy. I want to know why this is needed for enabling [company] ads in my app….

The same ethic of care appeared on XDA. But because of the heavy presence of skilled hobbyists who identified with devs, and vice versa, it was more often expressed as self-care: developer–hobbyists had privacy rights to protect. In a 2013 thread discussing the Facebook app, corp_corp wrote,

Why I disabled facebook app on my phone. Don’t want Facebook to track my location everytime I try it. We need a class action lawsuit because we have no choice to disable it …
Conversely, user privacy expectations were sometimes characterized as obstacles to development. This was particularly true on XDA, where sharp lines were drawn between expert technologists—hobbyist power users and developers—and the broader public. In a 2010 XDA thread about a face detection app, Davis2338 wrote,

> It’s not like we google faces. We know Google had the ability to do this with Goggles [sic, presumably Glass], but didn’t because of human privacy whiners.

User authorization for privacy could thus surface tensions between users and devs. In a 2012 thread about software to help devs “watch ppl on ur app,” iOS dev k*lee wrote, “This sounds good to devs, but, it obviously invades the privacy of users.” This conflict was not necessarily users’ fault. Some iOS devs felt that Apple assigned authority to users. In a 2011 discussion of an app rejected for privacy reasons, Carvellous wrote to the original poster, “That’s a valid point, but I doubt Apple is gonna budge, they care more about users than devs.”

Users were a complicated source of privacy authority. Sometimes they needed to be protected, and other times merely satisfied. Some devs felt that users did not care about privacy, and that notice and consent would only scare them. AS w7x posted in 2011,

> No one cares about privacy policies. They assume the data is already protected. Just saying “No personally identifiable information (PII) leaves the phone. This greatly reduces the risk of privacy problems” makes me wonder why I would have to worry about this to begin with.

The original poster responded to agree that a privacy notice might set off unnecessary alarm bells. But he went on to justify his PII notice through the authority of a specific group of users for whom the app was designed: lawyers and legal researchers.

_Platinum authority._ In the iOS forums, references to Apple’s authority were the single most-coded-for item. Examples frequently included privacy advice followed by a warning that noncompliance would trigger rejection from Apple’s store. As davezwarez advised in a 2009 thread, “You gotta ensure that the user knows and agrees to allow you to collect personal data or Apple may reject your app.” In a discussion about automatically subscribing customers who downloaded an app to an email list, iOS dev Funkicular put it memorably, “Whoa, this is soooo much worse than I thought. If Apple finds out what they did, they’ll get into a shit storm.” Subsequent commenters agreed with Funkicular, with two participants suggesting that the app in question should be reported to Apple. The dev community censured the app in question for perceived privacy violations on Apple’s authority.

It is difficult to overstate the importance of the App Store approval process in the discussions on iPhoneDevSDK. The longest thread on the forum is the App Store Wait thread. Active since September 2008, the thread had over 257 pages of replies at the time of writing, each cataloging and unpacking acceptances and rejections from the App Store. While devs often individualized their own App Store rejections as the bad judgment of a specific reviewer, devs as a whole recognized that Apple had an overarching privacy philosophy with which they had to comply. It was “Apple,” rather than individual reviewers,
who requested specific notice icons, or “Apple” that mandated user consent to tracking. In a 2012 discussion about getting rid of a location data awareness icon, Kairos warned the original poster that “Apple wants the icon there, so the user knows that the GPS is running, both for battery info and for privacy reasons. Getting rid of it is working against Apple’s design.” Kairos’ iPhoneDevSDK profile page showed over 9000 forum posts and a Seventh Anniversary Badge. He was a repeat participant in threads about privacy. His comments about Apple’s requirements carried weight not because he was particularly skilled coder—he admitted in the thread to never having worked with jailbroken iPhones—but because he had repeatedly tested the boundaries of Apple’s authority through successful app store submissions and could draw from those experiences.

Although Apple’s privacy guidelines were sometimes seen as protection—particularly from unscrupulous third-party advertisers whose software might corrupt otherwise privacy-sensitive designs—they could also serve as obstacles to development. A number of threads dealt with Apple’s decision to deprecate the unique device identifier (UDID) in 2011. This discussion highlighted a challenge Apple’s authority posed for developers: Apple’s guidelines could change over time. iOS devs were worried that new identifier solutions would eventually face regulation. In 2012, Jouissance offered,

I hashed the Mac address as an alternative [to UDID], and it offers the exact same uniqueness (and so suffers the risk of Apple fighting it later on).

Changes to app regulation troubled iOS devs because Apple’s authority was ultimate: a denied app was a waste of devs’ time and money. In a thread about rejections by Apple, silvioc noted that pivoting app designs based on Apple policy changes was a resource drain for small companies. Other devs in the thread were resigned to this state of affairs. As C@able posted in 2010, “we work with Apple. Apple is the boss. So when boss changed mind, then we should too.”

Despite the soft policy presence of Google’s developer guidelines, discussion of compliance with, or enforcement of, these guidelines was almost completely absent on XDA. Google was invoked to authorize privacy, but for reasons opposite of those invoked by iOS devs. Google was a figure from which users needed to protect their privacy. Devs could prevent Google’s privacy violations by building the right features. In a 2010 thread titled “Privacy Concerns—Google Firmware,” ImTheZoltan wrote,

At the core this is a pragmatic question: “How do I stop Android from collecting information about me?”... If you download Google apps, that’s a very different story. Those are closed source and probably do collect your data for ads. If that freaks you out, get a ROM without them.

A “ROM without them” references alternative operating systems—modifications built outside of Google—that come with basic apps such as an address book and calendar. ImTheZoltan suggested that these alternative systems could help concerned users avoid Google apps’ data collection. Apple authorized privacy as a design concern through its regulatory interventions, while Google authorized privacy through its reputation for collecting user data.
The figures authorizing privacy for devs were largely similar across platforms, with the major exception of the platforms themselves: Apple set the privacy rules for iOS devs, where Android devs saw Google’s behavior as justification for privacy design.

**Defining privacy in iOS**

After devs concluded that privacy was a legitimate design concern, they spent a large amount of time trying to delineate what counted as privacy in their products. While the justifications for privacy were similar across ecosystems, the ways in which privacy was defined and operationalized differed. In the iOS ecosystem, Apple’s gatekeeper status played a powerful role in defining the boundaries of privacy. iOS devs debated diverse definitions of privacy but often settled on instrumental definitions reinforced by Apple’s review process.

*Maximalist definitions.* Maximalist definitions of privacy, in which any collection of personal data raised concerns, were one form of privacy definition in iOS. As MouseTafa put it in 2009, “There is, still, the bigger ethical question of whether ANY transmission of ANY user data should be done without at least alerting the user.” MouseTafa’s point was made in a larger thread debating whether a user analytics package was “spyware.” He concluded that it was not, and that devs need not alert users to every piece of software used within their app. But the “bigger ethical question” shows that he and other devs consider maximalist privacy definitions, and understand their work to come in tensions with these definitions. “User data” in this case expanded beyond PII to include any information that could conceivably be collected about users. That devs debated maximalist definitions also hints at a recognition of conflicts between privacy and other design values, such as efficiency and profitability.

*Restrictions on data sharing.* Also present in iOS were privacy definitions centered on prohibitions against sharing data beyond the app company. Examples include promises that “… all data will be kept private,” and incorporated concerns about sharing data with third parties. In 2010, Alberf noted,

> My concern is, does Apple or other companies, like [SDK C] also collect user location including UDID [an identifier] at the same time when our own servers collect this info? If so, is it a privacy violation?

Restrictions on data sharing reflect an element of contextual approaches to privacy (Nissenbaum, 2009), as devs recognize that sharing data beyond an application may violate privacy expectations.

*Notice and consent.* The most frequently discussed privacy definition in iOS focused on transparency with users, particularly in the form of notice and consent. Devs frequently defined most kinds of data collection as allowable as long as users were informed. They frequently credited Apple with authorizing this particular definition. One 2013 thread captured both dynamics. The original poster asked for help designing an app that could
lock other apps out of location services, as well as track installation patterns. Forum veteran Kairos, introduced above, responded to the first query with, “Are you asking whether this app can lock other apps out of iPhone location services. The answer is NO, definitely not. Apple would never allow that.” Kairos responded to the second request with, “Are you asking if you can track the location of every user for your app? The answer is yes, with the user’s cooperation.” He continued with the stipulation, “You need a clear privacy policy that told the user that you were going to upload their location information, and let them know what you were going to do with that information.”

The different ways that iOS devs authorized and defined privacy illustrate a consensus over privacy as a concern but less consensus over what “counts” as privacy. Privacy functioned as an invisible fence for iOS devs: an obstacle structured by Apple’s approval process, whose dimensions were unclear, and which devs must feel their way around during development. Veteran devs drew on their experience to guide novices. By working together in the forums, iOS devs collectively mapped the boundaries of Apple’s invisible fence.

**Defining privacy in Android**

Lacking the regulatory “fence” imposed by Apple, Android developer discourse positioned privacy as a choice for consumers in a mobile ecosystem full of potential threats, including from Google itself. Android devs created a marketplace of privacy-enhancing applications, born of the loosely regulated open-source platform and the close developer–hobbyist relationships within the Android ecosystem. Two broad, related themes of approximately equal frequency surfaced in XDA: privacy defined as a technical feature, and privacy defined as user control.

*Privacy as a feature.* Android devs developed privacy-centric applications to distinguish their product within the marketplace, using privacy as a feature to set their applications apart. An example of this was a 2011 thread surveying forum participants on whether they would use a new privacy-enhancing application. Groln posted,

Hi, I have just developed some privacy protection software for Android. It blocks access for any installed app to the following data: [comprehensive list of data types including identifiers, GPS, call logs, browsers info]. For device ID, phone and mailbox #, SIM serial, subscriber ID and GPS it also lets you give custom or random values. Unlike similar apps, this doesn’t cause crashes when access to private data is blocked.

Responses were enthusiastic. WindUp wrote, “Would be v cool to use it. Its hard to keep privacy without not installing apps from the Market.” TvXY chimed in,

I’d totally use this app on my galaxy s and tab. Especially the feature where you give apps random or custom information instead of just blocking access—that’s crucial! If you want testing assistance those mentioned devices just let me know. Fingers crossed you get the positive feedback to port and keep building this thing.

TvXY’s response highlights a technical privacy feature—fake data to fulfill app’s data requests—and the enthusiastic uptake of that feature by skilled hobbyists.
TvXY’s response also recalls a privacy definition found in the iOS ecosystem: restrictions on sharing data with third parties. Frequently, members would tout services that did not require third-party data hosting or cloud-based services like VamOOS in 2010:

The issue with [text-messaging app] is its 3rd party server so if ur worried about privacy, then [competing app] is far stronger. I use [competing app] and … it’s a direct connection so no privacy concerns. Very solid.

Similarly, the creator of the “Where You At” location-sharing app promised in a 2014 thread to update “Privacy & Stuff,” particularly by ensuring that location data would not be stored or shared with third parties. Respondents analyzed his source code, pointed out vulnerabilities, and collaborated on solutions. The commonalities of data-sharing prohibitions in both ecosystems suggest that this definition might provide inroads for promoting more complex, contextual notions of privacy.

**Privacy as user control.** Another prominent privacy definition in Android was of privacy as individual control over personal data. Products would highlight their encryption features, for example, emphasizing affordances for user control. As dev 67 others posted highlighting their new app in 2013, “Fynd uses end-to-end encryption to ensure that your location data is only seen by those you publish it for.” Android devs and hobbyists often emphasized how important it was for privacy-sensitive users to maintain control over the full mobile stack, from hardware to software, to signals. These lessons were full of cautionary tales regarding hackers, Google’s data collection, or NSA surveillance. Defining privacy as user control over data led to another class of privacy-preserving applications: apps to protect *interpersonal* privacy. Applications like Blockage and Blockage+ (described in a 2014 thread) locked everything but the phone’s call dialer phone, blocking friends borrowing the phone from snooping through other applications.

Defining privacy as user control led to a wealth of privacy-enhancing options within the Android ecosystem. But privacy-enhanced applications were often developed by hobbyists and open-source developers working on their own. This led to a marketplace with many choices for consumers, but little indication of the level of professionalization, trustworthiness, or product support. Evaluating the privacy-enhanced applications discussed in XDA required a high level of technological literacy. In a 2011 thread about ad blockers, participant ExDot wrote,

xda is huge across the web but still a sort of “niche.” only a certain slice of those us who like doing stuff to our handsets fit in. suspect most of the market has no clue what even rooting is.

Others recognized the problems of Android’s free market approach to privacy and security. In a thread about threats to Android security, a participant railed against Apple’s strict app store model. Carboncopy responded, describing “the wild west that is Marketplace” and wrote,

Totally agree with you on Apple, that’s a big part of why I picked a Desire over an iPhone, but the Android approach is too far the other way IMHO. My two cents, be hopefully maybe just
maybe someone at Google is reading and thinking: hey know what, it is an accident waiting to happen.

Carboncopy appeared to be in a minority among XDA participants, however. Most placed responsibility for privacy protection on users. In a 2012 thread called “App Privacy Violation Concerns,” coldbear expressed concerns about apps that “start off w kosher permissions” but gradually add permissions as they update over time. Zachrospesiak responded,

Sure but the [Android] market does not auto-update the app if its permissions have changed which is really the best you can ask of Google. They COULD do the apple thing and filter every app that gets submitted, but I for one (As a dev making apps) would be seriously pissed if they did so, apples submission process can take 3 weeks while google takes at worst 1 hour to appear in the Market […] Google did all they can, at the end of the day its up to the user to read before installing and not just clicking around, its there fault if they install apps with more permissions than they really needs

For devs and skilled hobbyists, Android enabled access to privacy-enhancing applications, limited only by skill and literacy. The charms—but also the underlying inaccessibility—of Android privacy were summed up by mavenz in a 2013 thread about privacy problems in the Facebook app: “Whatevs. This is why i < 3 android. I can just hack something better.”

**Discussion**

Analysis of privacy discourse in iOS and Android forums illustrates that the ecosystem in which personal data are collected, processed, and shared matters to how data collectors legitimize privacy and how they ultimately define and implement privacy. Previous research has shown stark differences between Android and iOS apps with respect to privacy best practices (Zang et al., 2015). Our research helps us understand why. iOS privacy discussions illustrate that platforms can be powerful regulators. Apple’s definitions of privacy (primarily focused on notice and consent) are widely adapted and used by devs in the Apple ecosystem. Android’s free market model inspired less frequent privacy discussions, but the flexibility of open-source development encouraged a range of innovative, if sometimes inaccessible, privacy features to flourish. Critics have pointed to the unregulated Android store as a nexus of privacy problems in Android (Zang et al., 2015), and that assumption holds. But the story goes deeper: Google and Apple’s governance of mobile app development occurs not just at the App Store submission checkpoint but also in platform feedback (or lack thereof) received by developers, the community’s negotiation of the meaning of that feedback, the training of developers through online materials and offline conferences, and the lessons ingrained in operating systems updates and permissions protocols. Fundamentally different development cultures emerge on each platform, with different results for consumer privacy. The “wild west” of Android development means that privacy solutions abound for skilled hobbyists but that baseline privacy measures for the masses are lacking.
Several challenges emerge from these findings. First, iOS devs seem to be more interested in finding the acceptable boundaries of Apple’s definition of privacy than they are in exploring more complex meanings of the term. The current privacy requirements of the App Store narrow devs’ privacy imaginaries. This is problematic because privacy is a more complex notion than reflected in Apple’s current policy guidance. Apple policies are based on what many privacy scholars argue are outdated notions of notice and consent (Martin, 2013). Missing from both Apple’s regulations and, importantly, dev conversations are discussion of alternate privacy models, such as privacy as negotiation of identity boundaries (Cohen, 2012) or privacy as contextual integrity (Nissenbaum, 2009). While these definitions of privacy are increasingly important in research and law, they are not widely discussed or debated among developers.

Platforms might use their power as regulators to address this challenge by introducing new (and we would argue, better) definitions of privacy into their ecosystems. Recent news coverage and the WWDC event discussed in the “Introduction” indicate that Apple, for example, hopes to encourage differential privacy (a technique that keeps individual user data inaccessible while enabling insights from group behavior (Dwork, 2008)) as a new design norm within its ecosystem (Simonite, 2016). Our research indicates that Apple may successfully encourage differential privacy through either its policies or the technical restrictions it places on its developers. If required to do so, devs will scramble to figure out how to operationalize differential privacy in a way that gets them to market without compromising the data collection practices on which their profits often rest.

It seems unlikely that Google would choose to legislate a shift in privacy definitions. However, Google can indirectly encourage innovative privacy features to become Android market choices by finding ways to make new forms of privacy protection (such as differential privacy) technically accessible to their developers. But the “wild west” of the Android marketplace (as described by participants) raises the difficult issue of trust. How do consumers know to trust the promises of the privacy applications they are using? As Carboncopy put it in 2010, “I have a choice right now: Trust an app or refuse to use it. Simple, see? No other choice.”

Additionally, XDA developers were frequently not very reassuring when questioned about trust by their peers. In 2013, a dev who built an alternative to the Facebook app without location tracking was asked by R3D3: “I gotta ask, how can we know you didn’t mod this [app] so our info gets re-routed back to you?” The app’s creator, q2rex, replied, “bc im 1st: not that smart;). 2nd, Wayyyyy too lazy to work that out:P Trust me, i got better stuff to do with my time.”

A final challenge is a lack of transparency in how platforms regulate privacy. The pervasive feeling among iOS devs that privacy was an always-moving target, and the notable energy spent trying to find the invisible fence, reflects this lack of transparency. On the other hand, Apple can act more quickly than governments. The power of Apple as a privacy regulator links to calls to better understand the politics of platforms (Gillespie, 2010).

Conclusion

Mobile apps are a critical medium through which sensitive user data circulates. Thus, the values embedded in their design are an important site for research and advocacy. Because
Android and iOS apps (to differing degrees) are purchased in centralized marketplaces and compete for user attention with thousands of similar apps, developers must adapt their design values to fit those of the platform on which they build. The political struggle to define how privacy works in the mobile ecosystem is often concluded before consumers pick an app. That struggle takes place in a thousand tiny interactions among devs, hobbyists, the platform, and third-party vendors. The nature of the struggle and its results differ between platforms, but the struggle is always bounded, and those boundaries are policed and adjusted by the platform and its representatives.

Mobile platforms are not just passive intermediaries supporting other technologies. Rather, platforms govern design by promoting particular ways of doing privacy, training devs on those practices, and (to varying degrees) rewarding or punishing them based on their performance. Unique technocultures emerge in the iOS and Android development ecosystems, each producing a different set of development practices for privacy designs. “Privacy by design” is thus perhaps better termed, at least in the mobile development space, “privacy by permitted design” or “privacy by platform.” For privacy advocates looking to educate—or sanction—mobile devs, the power of platforms must be taken into account. Their influence manifests not only within the technical and policy specifications of the platform itself but also in devs’ interaction with platform representatives and each other. State regulators should similarly recognize the power of the platform to define the boundaries of privacy within their ecosystem. And we hope other researchers build on our work to explore the power of platforms to bound not only the limits of acceptable speech in the public sphere but also the limits of acceptable labor and design practices supporting that public sphere.

Acknowledgements

This work has benefited from discussions at the 2016 iConference and the 2016 Privacy Law Scholars Conference. Tarleton Gillespie provided valuable feedback on a more developed draft that emerged from those conferences.

Declaration of conflicting interests

Google did not approve or influence the results of this study.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship and/or publication of this article: This study was funded by the US National Science Foundation awards CNS-1452854, SES-1449351, and a Google Faculty Research Award (Shilton has received research grants from Google).

Note

1. In the more restrictive iOS context, where rooting is more difficult, less common, and more strictly policed by Apple, this is referred to as “jailbreaking” a phone.

References


**Author biographies**

Daniel Greene is a Postdoctoral Researcher with Microsoft Research New England. His research explores how we learn how to work in the information economy, with a particular focus on the institutions that link technologies, values, and the labor market.

Katie Shilton is an Associate Professor at the University of Maryland's iSchool, where she leads the Ethics & Values in Design (EViD) Lab. Her research explores ethics and policy for the design of information collections, systems and technologies.