SM+S
social media + society

Multi-Situated App Studies: Methods and Propositions

Michael Dieter[1], Carolin Gerlitz[2], Anne Helmond[3] (ID),
Nathaniel Tkacz[1], Fernando N. van der Vlist[2] (ID),
and Esther Weltevrede[3] (alphabetical)

Abstract

This article discusses methodological approaches to app studies, focusing on their embeddedness and situatedness within multiple infrastructural settings. Our approach involves close attention to the multivalent affordances of apps as software packages, particularly their capacity to enter into diverse groupings and relations depending on different infrastructural situations. The changing situations they evoke and participate in, accordingly, make apps visible and accountable in a variety of unique ways. Therefore, engaging with and even staging these situations allows for political-economic, social, and cultural dynamics associated with apps and their infrastructures to be investigated through a style of research we describe as multi-situated app studies. This article offers an overview of four different entry points of enquiry that are exemplary of this multi-situated approach, focusing on app stores, app interfaces, app packages, and app connections. We conclude with nine propositions that develop out of these studies as prompts for further research.

Keywords

app studies, mobile apps, situatedness, app stores, software studies, infrastructure

Introduction: A Decade of App Economies

10 July 2018 marked the 10-year anniversary of Apple's App Store—a significant milestone in the political economy of software cultures. From the beginning, the App Store contained not only official or "first-party" applications developed by Apple but also apps published by third-party developers. Shortly thereafter, on 22 October 2008, Google launched Android Market—which was later rebranded as Google Play—and similarly made apps available for the Android operating system. Contrary to the web, which was originally imagined as a shared information space (Berners-Lee, 1996) and only later turned into a commodified space, apps were conceived as informational commodities from their inception (Daubs & Manzerolle, 2015; Morris & Elkins, 2015; Nieborg, 2015). Today, mobile apps have become significant cultural and economic forms (Miller & Matviyenko, 2014). As of May 2018, Google and Apple own the leading app stores worldwide; there are over 3.8 million Android apps and 2 million iOS apps that generate over USD 86 billion in revenue. With the average user spending almost 1.5 months per year using them, apps have become deeply embedded in our everyday lives (App Annie, 2018). Apps are "mundane software" not only because they support everyday practices but because they insinuate themselves into our routines and habits (Morris & Elkins, 2015).

Despite their growing importance, apps pose empirical challenges for media research because of their tendency to move into the background while remaining thoroughly entangled with data-intensive infrastructures and the economic models of platforms. Apps are designed to perform as concrete software objects but are continually transformed—what Zittrain (2009) has conceptualized as "contingently generative"—through interactions with diverse socio-technical situations. From app stores to our personal devices, apps are transmitted as packages through seamless, highly automated downloading and purchase procedures and organized

[1]The University of Warwick, UK
[2]University of Siegen, Germany
[3]University of Amsterdam, The Netherlands

Corresponding Author:
Anne Helmond, Media Studies, University of Amsterdam, Turfdraagsterpad 9, Amsterdam 1012 XT, The Netherlands.
Email: A.Helmond@uva.nl

market layouts of stores to icon grids on our devices. For users, they seem like fixed objects; we drag apps around, bundle them into folders, or activate them by pressing their icons. However, the notion of apps as entirely self-contained also belies their involvement in the data flows of multi-sided platforms and their necessary entanglement with varying hardware devices and digital infrastructures that make their operations at once possible and, indeed, valuable.

The bounded appearance of apps is achieved through the specific ways their identity is regulated by software engineering. In the terms of Hui (2016), apps are "new industrial objects" consisting of logical statements and structures whose associated milieux includes algorithms, databases, and network protocols. The file format of the app—the "package"—delivers an abstract coherence that nevertheless can be decompiled, recombined, and reassembled in different ways. App packages can be understood on vertical scales, which might include metadata, folder hierarchies, and nested information, but also on horizontal scales by relations such as HTTP network connections, trackers, or platform integrations. The discreteness of an app is a technical achievement that is woven throughout the entire object but also supports a certain kind of multivalence as it enables its stakeholders, such as app stores, developers, partners, and users, to integrate and valorize it in multiple, simultaneous situations. In other words, apps have built-in tendencies to be situated and to situate themselves within different operative situations: they are made available in particular ways that follow the principle of extensibility in software development while in turn rendering infrastructures, sensors, or networks available for themselves.

When we deal with the technical situatedness of apps, it is not just a question of objects or sites of research but also a question of concretized systems of relations in standardized infrastructures. One might draw a contrast, in this respect, with ethnographic approaches to global commodities that aim to "follow the thing" across multiple sites or locales (Marcus, 1995). Since apps exist as digital objects within a technical milieu, it is less a case of following an app across "sites" than of situating and re-situating apps drawing from a number of unique affordances that are available to the researcher. Indeed, the kinds of reciprocal causality initiated by apps are generally highly controllable due to their logical composition as statements and data structures (Hui, 2016, p. 56). Exploring the infrastructural situatedness of apps in different conditions or states can, accordingly, give rise to new possibilities for accountability, visibility, and knowledge. Even while occasionally encountering forms of resistance and constraints (as we will discuss), it is possible to tease out different app forms, values, and relations by staging experimental and exploratory situations.

In this article, we call this a multi-situated approach to apps, where each methodological orientation or "entry point" simultaneously deploys and makes visible different infrastructural settings. Therefore, we propose a socio-material

and methodological perspective to situations. Contrary to classical (micro-)sociological understandings of situations, which suggest that situations are characterized by corporeal co-presence and defined by the actors involved (Gießmann, Röhl, & Trischler, 2019; Goffman, 1981), we approach situations at multiple levels of scale. Situatedness, in our understanding, includes the common understanding as "the involvement of the researcher within a research site" (Vannini, 2008, p. 815), similar to the way that app use practices and technical agencies are embedded in various infrastructures. This perspective invites us to reconsider situations as being scalable and connected (Nicolini, 2017) and to traverse between the micro- and macro-practices and infrastructure. Therefore, our methodological propositions both stress and employ the socio-material situational embeddedness of apps in multiple ways.

In what follows, we present four methodological entry points—app stores, app interfaces, app packages, and app connections—through which researchers can actively invoke different app situations and use these to advance or initiate an enquiry. This may entail the performative usage of an app through purposefully orchestrated, personalized user experiences, but it may also involve using data flows originally designed for machine reading. The selection of entry points draws on key sites within which many stakeholders engage with apps but renders them in specific ways to create research situations, taking inspiration from but also noting the challenges this poses for software and platform research. The final part of the article reflects on these challenges in the form of nine propositions for situated app studies.

Methodological Entry Points for App Studies

App Stores

A key entry point to study the situatedness of apps is in stores such as Google Play and Apple's App Store as well as country-specific and device-specific app stores. App stores are the main site for accessing, downloading, and distributing apps, and they allow researchers several opportunities to follow the perspectives of different stakeholder groups, including users and developers. In this way, app stores function as key gatekeepers—or as "obligatory passage points" (Callon, 1984; Fagerjord, 2015)—by setting up the rules for app creation, sorting, and distribution, and they do so by drawing from the economic model of the multi-sided marketplace (Rochet & Tirole, 2003) to ensure app exposure to potential customers. To account for the research affordances (Weltevrede, 2016) of app stores, we need to understand the specific situations that app stores create for apps. Indeed, they allow for a multi-dimensional perspective on the relations between apps as organized by the assemblage of stores as well as their algorithms, rankings, and developer and user activities. App stores can accordingly be used to address relations among apps as an application

of "issue mapping" (Marres, 2015) in platform studies of developer engagement and innovation, media concentration, and the conditions of possibility for practices.

It is important to note that while the two dominant app stores are the main focus point of this article, there is an abundance of app stores and repositories. There are manufacturer-specific stores (e.g., Samsung Galaxy Apps, BlackBerry World), country-specific stores (e.g., Yandex. Store in Russia, Tencent App Store in China), stores by telecom operators, and dedicated open source and adult stores ("App Stores as Data Infrastructure," 2013). Each app store comes with its own affordances, built-in logics, and mechanisms; thus, each store favors certain kinds of research over others due to its design, interface language, and user and developer communities. While app stores are often considered a relatively new phenomenon, they have been around for much longer in the form of software libraries, distribution platforms, repositories, game stores, package managers, and so forth (cf. Morris & Elkins, 2015). The proliferation of app stores should be seen in a longer history of software fragmentation for different devices, operating systems, and world regions, leading to the creation of distinct archival databases, platforms, and marketplaces (cf. Basole & Karla, 2011). The open-source Android operating system, for instance, allows third-party developers to build their own alternative, non-official app stores for Android applications. The multiplication and relevance of app stores has led to the development of various third-party market insight companies such as App Annie, which aggregates data on app stores, app usage and markets, as well as historical data on app store search results and the most downloaded apps over time (Vonderau, 2018). Such companies also offer general guidance for market research and app store optimization services and therefore provide relevant sources for considering how app developers write themselves into the calculative processes of stores by strategically selecting categories, drafting descriptions, and presenting apps to make them store-ready. These third-party indices offer alternative, often aggregated access points to app stores by placing apps and stores in distinct commercial contexts.

App stores themselves come with different access points, as some offer web interfaces (e.g., Google Play), while others have limited web functionality (e.g., Apple's App Store) or have only device-based mobile interfaces (e.g., Yandex. Store). In addition, most app stores do not offer systematic access to their data via standard application programming interfaces (APIs), with Apple's App Store being one exception. This means that there are varying capacities for systematic search and data extraction (e.g., via API calls, web scraping, and manual retrieval). App stores allow for multiple ways of querying or exploring the offered app spaces, such as by searching for specific apps (e.g., [Facebook]), topics (e.g., [pregnancy]), or genres (e.g., [messenger]) or by browsing app categories (e.g., games), ranked lists (e.g., "Top Charts"), or featured lists (e.g., "Editors' Choice"). App

stores offer both algorithmic and curational ordering practices, which provide research opportunities to explore app collections based on the demarcation of the store and to follow possible user pathways. Users are also presented with additional app groupings on individual app pages. For example, Google Play has personalized recommendations ("Recommended for you," "You Might Also Like") and complementary app recommendations ("Related to this app"), but it also suggests recommendations based on topics associated with apps ("Similar Apps"). In contrast, Apple's App Store has sections called "More By This Developer" and "You May Also Like" (technically specified as "customers-also-bought"). A key challenge for working with "app relatedness" is the personalization and localization effects of app stores—similar to search engine research—which can determine their results based on location, country, and language but also previous user behavior. Central to the algorithmic ordering of apps are the categories that developers can assign to their apps, which inform similarity calculations but also users' engagement with apps by browsing these categories. In addition, app stores' categories and lists can be critically examined to provide insights into the built-in logics and mechanisms driving these categorizations.

App stores enable researchers not only to draw on but also to reconfigure these various set-making capacities of app stores. They can be employed as indices of apps that may be queried for keywords, genres, or developer names, allowing the device to demarcate a sample. One can consider which results are returned and where they are ranked across "spheres," per query, per country, and per store (Rogers, 2013, p. 118). As indices of apps, app stores typically provide individual pages with details about app titles, functionality, version, developer, screenshots, descriptions, permissions requested, download statistics, reviews, and ratings. App stores are thus key sites for studying how users, developers, and other actors influence the production, distribution, and reception of apps and app cultures (Morris & Murray, 2018). For example, app stores can be used to study the work of developers using app details, descriptions, and update logs or the reception of apps through ratings and reviews. Using the DMI Google Play Similar Apps and iTunes Store tools, we can extract the details of individual apps, collect "Similar Apps," and extract their details, as well. Such data can then be used to identify networks of app relations as created by the store.

Queries can be deployed to engage with collections of apps as opposed to focusing on single apps in isolation. First, stores can be used to generate thematic and issue-oriented collections that can be studied as expressions or indicators of cultural differences. For example, how are apps employed in relation to religion (e.g., Buddhism, Christianity, Hinduism, Islam, and Judaism)? ("Digital Methods for App Analysis," 2015) What solutions do app stores—as indices of apps—generate or recommend when users query for controversial, partisan, or objectionable content or sensitive issues (e.g.,

abortion, gun control, porn, or mental disorders and conditions)? ("App Stores and Their Bias," 2018). The latter, in particular, can be understood as an application of issue mapping (Marres, 2015) within the organizational logic and structures of the app stores themselves.

Second, the same approach can be used to explore different genres of apps and the practices they enable (e.g., health and messaging apps). Here, app titles and descriptions offer further ways to study app features, providing insights into key functionality and interoperability with other apps or platforms. How do different secure messaging apps position themselves in their self-descriptions, and to what extent do they address issues around security, encryption, and usability, for example? ("Mapping (Secure) Messaging App Ecologies," 2016). App descriptions can be used as starting point for manual categorization or for topic modeling through natural language processing, or they can be queried for predefined terms and phrases.

Third, a large proportion of apps are not created as stand-alone objects (only) but are built on, or in relation to, other apps, software, or platforms, for instance, by drawing on the APIs of platforms for data extraction/input or offering support practices for platform engagement. As platforms explicitly invite and facilitate developer engagement with their functionalities and data (Bodle, 2011), app stores can be used to explore how developers have built on platforms and how they intensify, support, interpret, alter, or amend their features, data, and associated practices (Gerlitz, Helmond, van der Vlist, & Weltevrede, 2016).

The app-sorting processes can themselves be the subject of empirical enquiry, opening up questions about how app relatedness is produced in the first place and how algorithmic sorting compares across different app stores over time. Although algorithmic processes are generally difficult to account for and interpret, it is possible to log search results and their rankings as they unfold over time. A comparative study of ranking volatility for selected queries across Google Play and Apple's App Store can yield important insights into their ranking mechanisms, issues related to media concentration, and the curation and removal of certain apps by app stores ("App Stores and Their Bias," 2018). Moreover, such approaches address not only ranking algorithms but also "ranking cultures"—highlighting the "distributed and heterogeneous agencies that converge" in ranked lists (Rieder, Matamoros-Fernández, & Coromina, 2018, p. 54). Such agencies can, for example, include aggregated types of user engagement with apps (e.g., downloads, ratings, reviews) as well as app store optimization tactics that are incorporated in these algorithmic rankings. Moreover, the app store can initially be used as an entry point to study apps to scope the app store itself, its ranking cultures, and contents and then to examine the cultural implications of these processes for the production, distribution, discovery, and experience of apps within the app store and beyond.

App Interfaces

While app stores support research on the relations between apps and their economization, app interfaces offer entry points for specific enquiries into the conditions of possibilities for user practices. Enquiries into interfaces can tell us not only about the apps but also about the expectations that those interfaces have of users and how certain ideas about users are designed into those apps. It has been a long-standing claim of science and technology studies (STS) of human–computer interaction (HCI) that shaping the user is a central concern of interface design (Woolgar, 1990), particularly through forms of embedded and enacted scripting (Akrich, 1992; Suchman, 2007). This kind of technical scripting raises questions concerning the circulation of power, subjectivation, and the production of value, which are especially pertinent under conditions of platformization and the data-intensive forms of "controlled consumption" that apps facilitate (Andersen & Pold, 2018).

The walkthrough is a method that can be used to explore interfaces, including how they "script" the user, by systematically documenting and abstracting interface features from their normative infrastructural settings. The walkthrough method is commonly used in software engineering and user-centered design research to present a software product to peers or stakeholders for review. It is also used in commercial technology reviews and has a longer history in product demonstrations and infomercials. Light, Burgess, and Duguay (2016) have suggested, however, that walkthroughs can be repurposed in a "significant departure" from these prior uses to perform a critical analysis of a given app. For these authors, the core of the method involves "step-by-step observation and documentation of an app's screens, features and flows of activity," all of which can be contextualized within the app's vision, operating model and governance, or what they call the app's "environment of expected use" (pp. 881-886). In this way, Light et al. propose reappropriating this method within a cultural study and STS framework as a contribution to the methodological study of apps.

In terms of strengthening the interdisciplinary context for walkthroughs, however, the method additionally benefits from a critical understanding of user experience design epistemologies and practices. This can, for instance, draw attention to the rise of behavioral design regimes as a mode of scripting the user that targets the nonconscious dimensions of cognition (Hayles, 2017), leveraging insights from behavioral economics, cognitive psychology, and neurological research while operationally relying on big data and nudging (Yeung, 2017). In this design regime, user journeys contain a series of key performance indicators that are "sunk" into interfaces as an environment to facilitate the captivation of the user. An emphasis on these characteristics benefits from forms of critical design literacy. Indeed, behavioral design or dark pattern libraries (Dieter, 2015; Nodder, 2013) might be

consulted in this respect to guide the analysis of formal components based on plotting user decision-making and actions.

It is important to recognize how walkthroughs are uniquely performative as a situated rendition of a user journey that foregrounds material characteristics of the interface. In this respect, the walkthrough is a methodological intervention that inevitably involves a user persona to facilitate the process of engagement within an app. Personas have been a mainstay of HCI and interaction design as a method that produces a realist fiction of a user (Cooper, 2004). It usually involves performing empirical research on a market or audience for a product and then imagining an ideal type that can be utilized to develop software or a service. Personas can be used to orient a situated enactment of a walkthrough—including processes of repetitive or habitual use that might be required to investigate personalization—but can also be used simply for practical purposes of connecting to other existing profiles in social media. Here, we might consider distinguishing between a user persona and a *research persona* that recreates abstract use case scenarios that are aligned with the interests of the researcher. Depending on the app, the creation of personas may require different degrees of emulation of native use situations, including material devices, location, and activity history, while other apps allow for more static walkthroughs. Personas can also be created to explore how technological infrastructures respond to specific practices and user identities, including non-use and attempts to disconnect (Baumer, Ames, Burrell, Brubaker, & Dourish, 2015; Karppi, 2018).

While the list is in no way exhaustive, we foresee a number of deployments of the walkthrough method for app studies. As Light et al. note, walkthroughs can be enacted through different phases of use, including signing in, everyday scenarios of routine use, and quitting an app. However, we see further opportunities arising from moving from single to comparative walkthrough analysis. One can consider, for example, how different apps handle key moments of the walkthrough (login, terms and conditions, support screens, verification), how specific features are organized (action points, input buttons, notifications), how design patterns are implemented, or how navigation paths are arranged. To illustrate, Figure 1 compares the different sign-up options across mindfulness apps as they appear in different stages of the walkthrough.

Comparative walkthroughs can also be used to contrast the multi-sidedness of platforms. Here, the approach involves adopting platform-afforded research personas—user, developer, advertiser, and so on—to consider how platform providers address their varying groups on different sides via distinct interfaces (cf. Bucher & Helmond, 2018). By de-emphasizing the user-centered app walkthrough, multi-sided walkthroughs can make visible economic and value creation strategies that are not apparent from the user side of the market. This, in turn, may open up a number of political-economic areas of enquiry, such as how apps reconfigure and regulate platform labor on the level of interface design. Finally, walkthroughs can be used for historical analysis, where versions of an app are considered to detect design, feature, or data capture changes over time. This might be run in an emulator or within environments such as Android Studio, including specific simulations of period hardware and operating systems.

In dialogue with these approaches, there are additional opportunities to refine the scope of the walkthrough, particularly in attending to the more formal or templated aspects that allow for data input (Gehl, 2014), including touchpoints, buttons, and forms that might be emphasized to indicate platform relations, such as the presence of Facebook or Google logins, which speak to the techno-economic relations of platformization (Helmond, 2015). As an example, Figure 2 contains a series of comparative walkthroughs of dating apps that explore how they are situated within specific data infrastructures ("Dat[a]ing: Mapping Data Infrastructures of the Dating Industry," 2017). In this case, the main concern involves tracing the magnitude and pacing of inbound and outbound data flows and linking the micro-practice of interface engagement with the distribution of personal information. The visualization strategy, accordingly, abandons the GUI-screen capture in favor of a more data-centric approach, enabling the quantification and comparison of informational disclosure patterns across a set of related apps.

Emphasizing the situatedness of app interfaces can, accordingly, be further developed through experimental visualizations to abstract shared features, data flows and infrastructural configurations, or forms of mapping that draw inspiration from techniques in information architecture. The latter might be utilized, for instance, to address particular challenges in the analysis of user journeys, such as through mapping branching paths or separate "support screens" to assist with priming the user for further acts of disclosure.

It may seem obvious, but it is important to stress that the walkthrough approach ultimately works with a series of interfaces. The screen captures used to document or annotate the walkthrough are, we suggest, not to be reduced to images and analyzed primarily through semiotic methods. Apps are first and foremost operational media; they are applications, things for doing. Importantly, apps are typically designed with behaviors—not meanings—in mind. App developers aim to get their users to do specific things—to change their behavior—and the walkthrough method can be used to reflect this behavioral focus. While user experience, usability, and cultural studies-inspired approaches place the user at the center of the research, we see potential in using walkthroughs to examine how apps are infrastructurally situated by teasing out data flows, design affordances (e.g., ideal user types and practices), and platform mechanisms as their broader conditions of possibility. The research persona therefore plays a special role as the methodological user surrogate, enabling access to app interfaces while facilitating heterogeneous research situations.
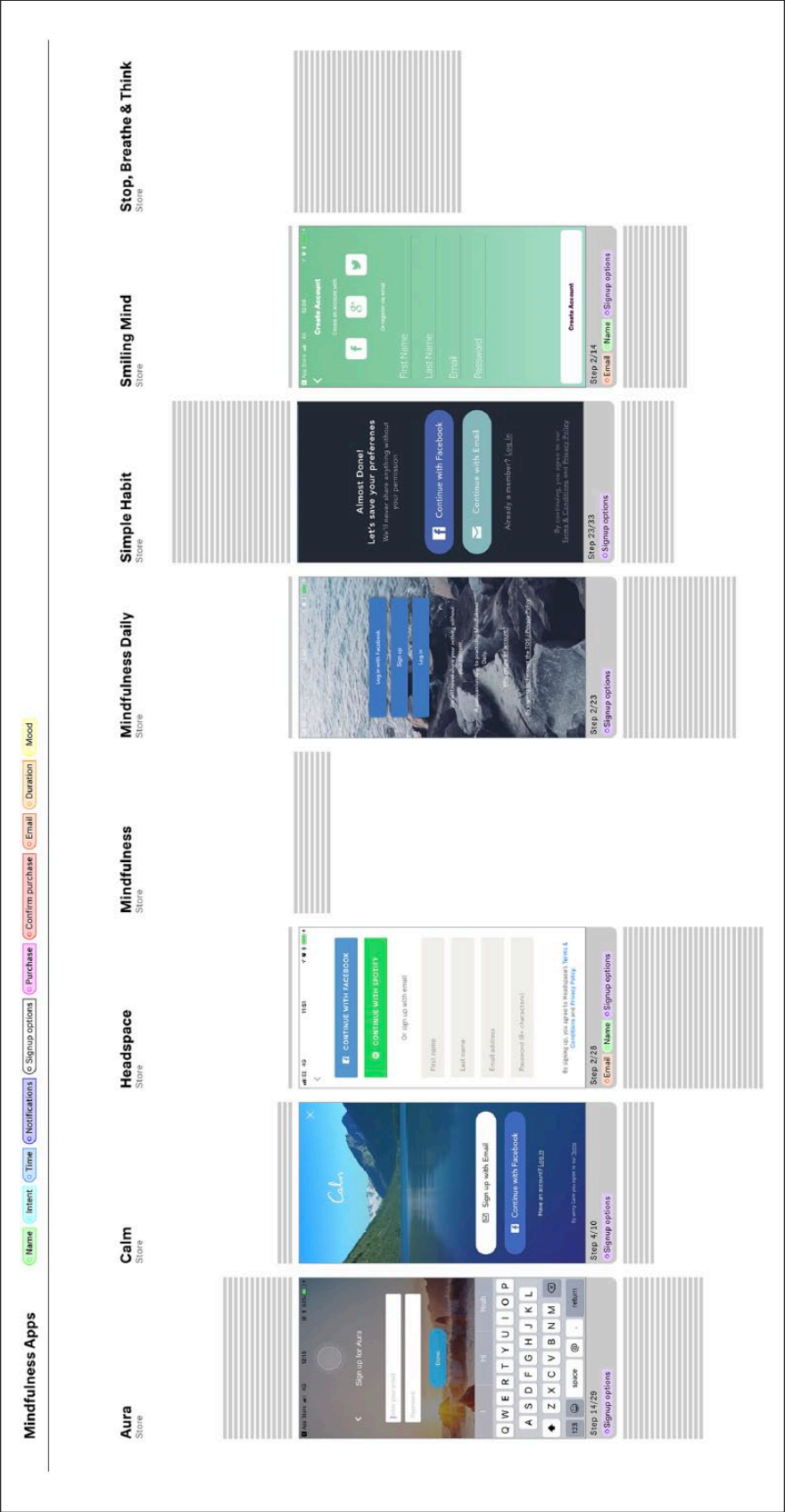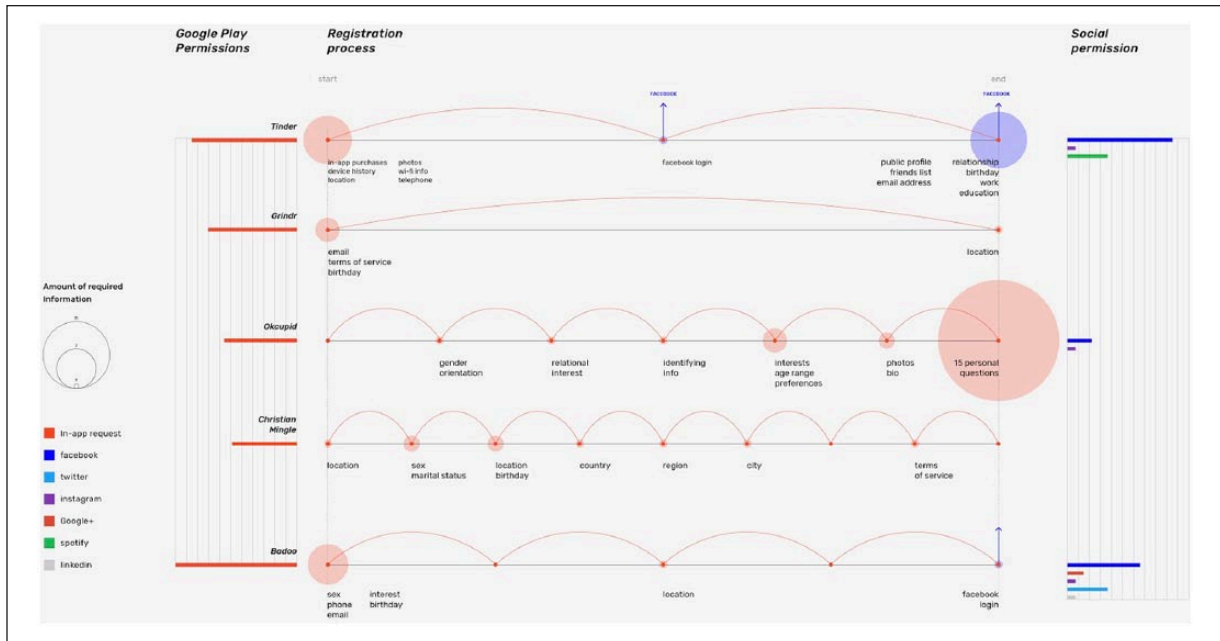
Figure 1. Side-by-side comparison of login or sign-up screens in a selection of mindfulness apps.

Figure 2. Comparative information visualization of the user registration process in dating apps (Tinder, Grindr, OkCupid, Christian Mingle, Badoo).

App Packages

Our third entry point allows for a more detailed exploration of the embedded infrastructural arrangements of apps by engaging with them as software packages, an experience that is usually shielded away from app users. Downloading and installing apps through official app stores is regularly presented as a seamless experience where users do not get to see the downloaded app on their devices; rather, they can only experience the automatically installed version. In this way, app stores obfuscate the status of apps as concrete software objects (cf. Morris & Elkins, 2015). To work against this obfuscation, it is necessary to re-situate apps outside their normative context of consumption by utilizing software repositories and tools for analysis as packages. First, however, let us briefly introduce the different software formats of apps.

The two main formats for mobile apps are .ipa (iOS application archive) files for iOS apps and .apk (Android package) files for Android apps. They are both specific types of archive files or compressed software packages that can be extracted to view the code and resources of apps. Developers upload these files to app stores, where they can be subjected to review before being admitted to the stores for further distribution. Users typically do not see these application archive files, as they are automatically downloaded and installed onto their devices in the background when they use app stores. Device manufacturers such as Apple strictly limit what can be done with their devices and only allow the installation of apps that have been approved by the official store. Downloading and installing iOS apps

outside of Apple's official App Store requires "jailbreaking" and unlocking the device as well as utilizing a third-party app manager such as Cydia or Cydia Impactor. Android, on the other hand, positions itself as an open platform; developers can distribute their apps via various third-party Android app stores and marketplaces or via their own websites. Users can then configure their device settings to download and install apps from unregistered developers outside Google Play. Such alternate marketplaces or websites, however, are still generally designed to facilitate downloading for standard use rather than to enable inspection of the software package itself.

Gaining access to an individual application archive file (as software) typically requires moving from official app stores to so-called app repositories or using other dedicated software. Much like app stores, app repositories are a storage location from which software packages may be retrieved and installed on a computer for personal or research purposes (Allix, Bissyandé, Klein, & Le Traon, 2016). These repositories are often presented as online directories of apps with download links to multiple prior versions of the package. While Cydia contains the largest repositories for iOS apps, the leading repositories for Android apps include Aptoide, APKPure, APKMirror, and F-Droid. App repositories may visually resemble the look and feel of official app stores and similarly display the most downloaded apps, app categories, and various search options. Repositories such as APKPure contain a wide array of apps but offer only a few versions of an app, while APKMirror appears narrower in scope but has many versions of the most popular apps. This suggests that

each repository is to be considered on its own terms; none operate as perfect archives, and each has different affordances for (historical) app research. Within software engineering, software repositories are an important source for studying the evolution of software (Kagdi, Collard, & Maletic, 2007). This immediately raises issues about running or emulating older APK packages to examine the larger techno-commercial ecosystems that apps are embedded in over time to or analyze the evolution of the app ecosystem at large (cf. Helmond, 2017). Working with app repositories also raises juridical concerns, as it is not always clear whether an app has been uploaded legally. Often, there can be issues with malware and spam in such app repositories. To prevent these concerns, it is possible to draw on dedicated software such as Raccoon to download current APKs directly from Google Play and to bypass the repositories altogether. However, Raccoon still requires authentication with a Google Account to retrieve apps and provides only the latest app version.

Situating apps as software packages allows research beyond the app's interface and into the code. For example, since APK files are always also valid .zip archive files, one can view their contents by unzipping the file. Other app packages such as IPA files may also be unzipped to view their package contents and structure, but these contents may be encrypted differently (e.g., due to digital rights management [DRM] restrictions). Some parts of apps may need further decoding to fully view their contents, and this can usually be achieved with the support of additional tools. Decoding apps shows the contents of the package, including all the necessary files and resources. The AndroidManifest.xml file, for example, describes metadata such as the name, version, and contents of the APK file and includes information about the app's permissions. As such, it can be used to examine how and where apps request and recombine heterogeneous data types, such as behavioral or sensor-based data (e.g., access to location) or user data (e.g., access to contact lists). This may open up questions into (the evolution of) app permissions in relation to public and policy debates about user privacy and data protection as well enquiries into the medium specificity of these recombined heterogeneous data flows as novel types of social data. Further resources in the package include software development kits (SDKs) used to build particular modules in the app, including social logins, app analytics, and advertising libraries, which enable research on the affordances of developer resources.

Re-situating apps as software also makes them available in addressing the range of enquiries found in critical code and software studies (Fuller, 2008; Montfort et al., 2012). One can study the code up close or parse it through other diagnostic tools enabling comparisons across apps or sets of apps, such as with Appcestry (Chao, 2018). Data sourced from files can also be used to complement other methods. For example, data on an app's permissions sourced from the AndroidManifest.xml file can complement a walkthrough

study of when and how an app collects user data. For the present purposes, app packages can be used to examine the various stakeholders involved in the production of apps—the *multi-sidedness* of apps—as well as the infrastructures for transferring and storing app content and data—their *multi-sitedness*. In particular, by analyzing inscribed libraries such as SDKs, it is possible to identify app templates and third parties embedded within apps that collect various forms of data, usually for advertising, authentication, and performance optimization purposes. We refer to these third parties as trackers.

While it is possible to examine APK files for trackers directly, we rely on Exodus Privacy—a "privacy auditing platform for Android applications"—to automate the process. Exodus scans APK files, compares them with their list of known tracking technologies, and generates reports for individual apps. Another tracker tool, the DMI App Tracker Tracker, is built on Exodus Privacy and extends its functionality to detect known tracking technologies or other software libraries in a set of APK files collected from an official app store or app repository. There is much related work in computer science and software engineering examining third-party libraries, including advertising libraries, in apps (Book, Pridgen, & Wallach, 2013; Ma, Wang, Guo, & Chen, 2016). To build a set or collection of apps, one can follow similar collection-making strategies as previously described for the app stores. The ability to make collections of apps for analysis is itself an affordance that is gained through such repositories.

Examining trackers embedded within mobile apps renders visible a number of otherwise obscured stakeholders. Comparative analysis of different apps or groupings of apps can also be used to determine which advertising or analytics providers dominate different areas or which types of apps are loaded with trackers. For example, one previous project examined tracker code presence within a number of app sets ("Mapping Data-Intensive App Infrastructures," 2018). In addition, with the help of repositories, it is possible to study how the presence of trackers in an app or group of apps has changed over time, which offers insights into changing app stakeholder relations and business model pivots or otherwise reflect dynamics in the wider economies of app advertising, app development, app analytics, and mobile game monetization.

App Connections

Our fourth entry point for multi-situated app studies is the network connections that mobile devices establish and that allow both multi-sidedness and multi-sitedness to be traced. These connections are often established on behalf of the apps running them and are needed for things such as user authentication, app updates, advertisements, and serving and uploading content. It is well known that mobile devices keep logs of the countless access points they probe while passing

through the access radius of Wi-Fi access points. Indeed, whenever location services are enabled, mobile devices attempt to connect to each and every access point that is listening for such access requests. As Mackenzie (2010) notes, the implication is that users' devices are continuously connecting to and disconnecting from objects and infrastructures "without knowing exactly how or where" (p. 5). When approaching apps as multi-situated in this sense, network connections are a key entry point for understanding how apps are always and necessarily bound up with—or "tethered" to (Zittrain, 2009)—other objects and infrastructures.

The proposed approach relies on methods from network security specialists (e.g., Enck et al., 2014), which are adapted to study the multi-situatedness of apps. To gain a sense of an app's connective entanglements with other objects, devices, infrastructures, and services, it is possible to capture and log the connections that are being established. Similar to desktop computers, there are many applications that monitor, track, analyze, and display inbound and outbound connections from and to a device. These applications typically sit somewhere "underneath" the application to capture the device's low-level network connections and hence might require privileged control (i.e., root access). While this is more common for advanced Android users, it is difficult for iOS users to "jailbreak" their iPhones. As a result, network connections cannot always be isolated or associated with the apps from which they were derived. Such apps typically log metadata such as dates and times, protocols, packet sizes, IP addresses, and bundle IDs of the apps connecting to these addresses. These details can be used to distinguish different kinds of connectivity (e.g., active or background, inbound or outbound), chart infrastructural relations to remote hosts and servers, authentication or authorization providers (e.g., OAuth, social logins), third-party content delivery networks (e.g., Akamai, Amazon CloudFront), cloud services (e.g., Amazon Web Services, Microsoft Azure), ad networks (e.g., AdMob, MoPub), and thousands of other tracking technologies. Each of these connections provides different affordances and insights into how apps are entangled with other objects and infrastructures to account for the multiple sites and sides of apps and to render the webs of connectivity that apps and mobile devices weave. In addition, they afford enquiries into user privacy, data protection, developer monetization strategies, and app-based revenue models.

Previous research on tracking technologies, cloud infrastructures, and data infrastructures is based mainly on web corpora ("The Tracker's Guide to the Cloud," 2014). However, some of these approaches may be resensitized to explore apps. For example, network connections can be studied to gain a sense of the many infrastructural relations, dependencies, data traffic flows, and third parties connected to apps. Once network connections are established and webs of connectivity are woven, they also serve as distribution channels to collect and deliver data traffic to anywhere

Figure 3. An encoded MoPub URL with unencrypted HTTP ad request parameters and values (device name, bundle ID, gender, age, lat long, screen width, height, language, carrier network, permissions).

between thousands and billions of mobile devices. While APK archive files can be employed to detect software libraries written into apps (e.g., SDKs) and thereby render static infrastructural relations, network connections and network traffic are ultimately dynamic and ephemeral infrastructural relations. They are established when an app is running—even when running invisibly in the background—but they are dropped as soon as the app is closed and cannot necessarily be rendered from an app's package contents. They are triggered by certain specific events, cues, or conditions that not all users or devices might meet, as in the case of loading personalized content or advertising, which poses challenges to approaches using source code analysis. Instead, apps anticipate users or user profiles to trigger these events or conditions, and the outcomes are specific to and dependent upon them.

The boundedness of apps as bundles or packages is challenged by the recognition that apps are routinely extending themselves through these network connections. On one hand, researchers can observe the topologies, rhythms, and volumes of inbound and outbound data traffic flows through network sniffing methods. For example, it is possible to detect and characterize different kinds of connections and implicated third parties for different sets of apps. Merely rendering the networks of third parties associated with apps visible is arguably a powerful rhetorical strategy for critical Internet infrastructure studies. In addition, there has been growing interest in studying the materiality of Internet infrastructure and signal traffic (Parks & Starosielski, 2015). What or whom do these connections serve? Furthermore, researchers can inspect or even intercept data packets transmitted insecurely across these network connections with packet inspection methods. Using common network data packet inspection tools such as Wireshark and tcpdump, we collected and analyzed query parameters in HTTP requests, which, among other things, yielded detailed ad requests to ad networks (Figure 3) ("Mapping Data-Intensive App Infrastructures," 2018). Such data are accessible through any unencrypted HTTP connection and may easily be captured.

Some network utility apps, such as Network Connections (Android), allow users to live capture network connections while using a device and to export the logs in standard tabular

file formats. Such features also point to the possibility to script certain uses, users, and use scenarios and to design research protocols that are more controlled or systematic. In these kinds of studies, it is paramount to craft the research and methodology carefully so that the situations elicited through them can be interpreted. One can use a "clean" researcher phone with "fresh" user accounts for the app(s) under study. However, one could also conceive of rich and mature profiles trained to enact a researcher's choreographed situation (e.g., to trigger cookies, personalization, localization, and targeted ads). Some of these strategies were originally developed for studying personalization and localization in search engine results (Feuz, Fuller, & Stalder, 2011; Rogers, 2013). Once network connection data are obtained through dedicated tools, it is sometimes also possible to trace them back to the firms and organizations behind them (Figure 4) ("Dat[a]ing: Mapping Data Infrastructures of the Dating Industry," 2017). Network connection data include IP addresses of the connections an app establishes, and these can be converted into domain names, locations, and ISPs of their hosts by using IP lookup tools. The results can be matched to other expert lists containing known infrastructural technology providers, such as Ghostery for web tracking technologies and CDNFinder for content delivery networks. To create network connection situations, questions to keep in mind are, what are the sites (locations, server hosts, data centers, cloud servers) that are being connected to? And which buttons—or what kind of scripts—trigger these infrastructural relations for serving content, serving ads, signing in, and sharing content?

Nine Propositions for Situated App Studies

The four methodological entry points introduced above enable the production of different situations within which the multi-sidedness and multi-sitedness of apps can be made available for research. At the same time, these entry points raise questions concerning how researchers in turn situate themselves methodologically toward apps and their socio-material relations. In what follows, we offer nine propositions to address the empirical and conceptual challenges of studying apps and similar digital objects:

1. *Move beyond ready-made social data.* Unlike the web or social media platforms, which are considered to offer a variety of user content and data ready-made for "social" investigations and accessible in structured ways via open APIs, apps are characterized by heterogeneous data formats ranging from prestructured interface-level data, sensor data, and network connections to software libraries and infrastructural data forms. Mobile apps may collect user content and data, but they rarely offer structured access via open APIs, which means there is an absence of app-based social data ready-made for researchers and

third-party developers. Apps that present social data through their interfaces (e.g., dating apps, chat, instant messaging apps) often cannot be easily retrieved, as the alternative web-data collection technique of screen-scraping is not accessible on mobile apps. Engaging with apps through the different entry points shows that data or features that may appear specific to one app may actually derive from a different data source, an imported plugin, device-based sensors, or external tracking or advertising technology. Apps invite researchers to reconsider what constitutes social data and to devise inventive methods that respond to the heterogeneity, the different kinds of structures, the various stakeholders involved, and the multiple origins of data.

2. *Navigate infrastructural resistance.* While obfuscation is a commonplace technique in computer science and software engineering (Matviyenko, Clough, & Galloway, 2015), the deliberate efforts or infrastructural effects that render code and data illegible to both human and technical interpretation create significant challenges for multi-situated app research. Some forms of obfuscation are intentional: APK files might be designed to frustrate decompiling tools; certificate pinning and encrypted channels limit the scope of packet sniffing; and entire app ecosystems, such as, Apple's iOS can appear essentially off-limits due to DRM protection. Others may result from the fact that functions or data flows do not address human users and are instead designed to be machine readable. In this sense, it is not uncommon for app studies to encounter considerable infrastructural resistance that might circumvent or side-track empirical research. Thus, despite recent claims that digital media open up new possibilities to empirically realize ANT's principle to "follow the actors" (Venturini, Jacomy, Meunier, & Latour, 2017), apps pose various challenges to the idea of following actors, data, or infrastructures. However, rather than seeing these challenges as black-boxed limits, it is more productive to consider instances of obfuscation as offering a spectrum of opportunities to navigate around resistances and re-situate apps to open up alternative (albeit partial) perspectives. Producing methodological situations, accordingly, may involve working both with and against the objectives and infrastructural resistance of apps.

3. *(Un)do the user.* While some strands of software, platform, and infrastructure studies bracket out users and their practices, such bracketing is not always achievable or desirable in studying apps, even when users and their practices are not the ostensible focus. Many apps require personalized logins, build on existing social media profiles, initiate data flows only through user practices, or are tied to users' specific
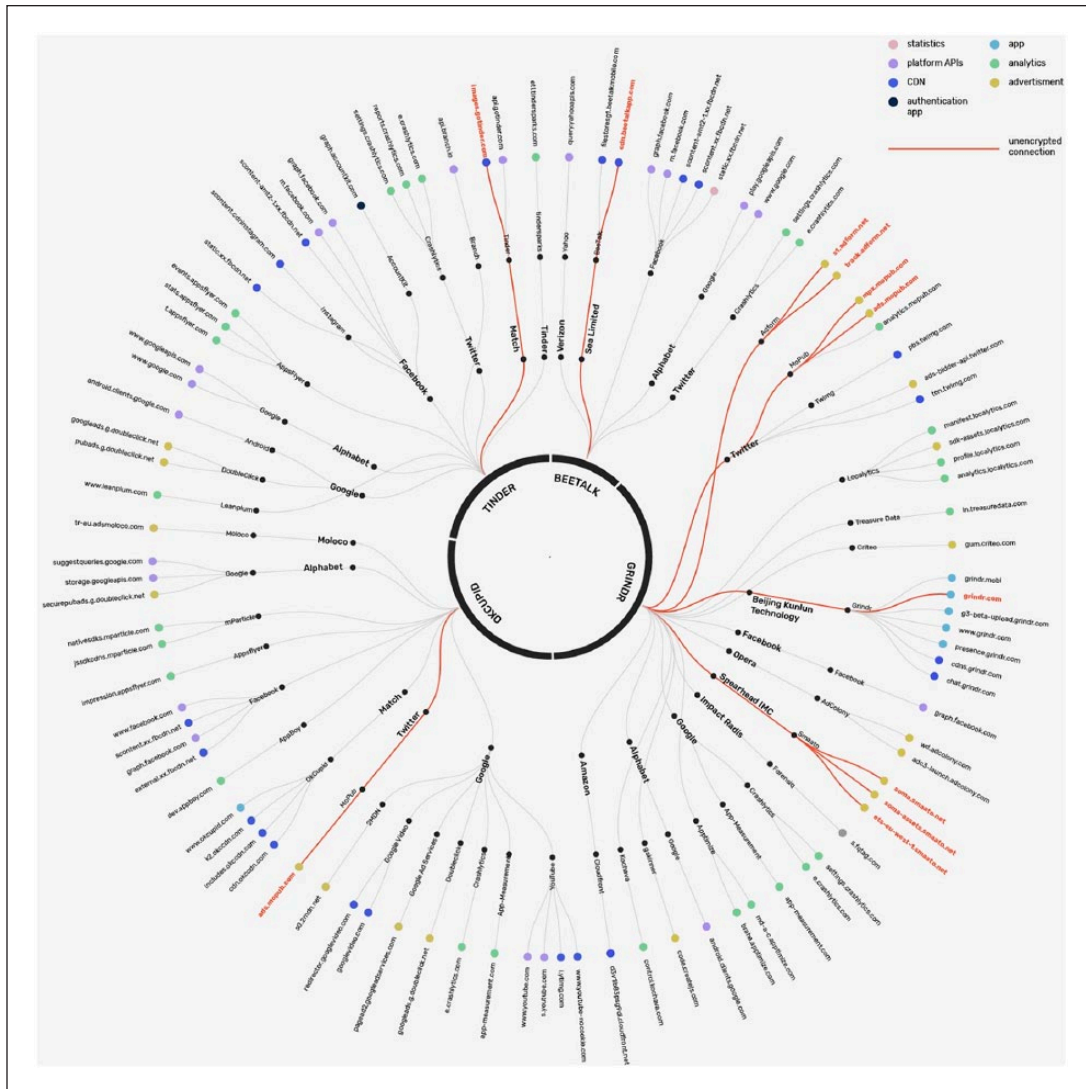
Figure 4. Companies behind the network connections coming in or out of four popular dating apps (OkCupid, Grindr, Tinder, BeeTalk).

locations. While disentanglement from some elements of user practices is possible in certain situations (e.g., minimizing the personalization effects of app stores) or is of little consequence in others (APK-based methods that work without profiles), other methods—such as walkthroughs and studying network connections or sensor-based data flows—are not possible without some kind of staging of the user experience. Apps present researchers with something of a paradox in that to make available non-user-centric enquiries such as those relating to infrastructure and political economy, they need to consciously re-create and pass through them. To mediate this issue, we suggest repurposing the design method of creating personas—fabricated users with habits, preferences, psychographic, or demographic specificities to inform decision-making around product development. Personas can be used to generate research situations by acting as kind of surrogate user. They can be deployed to create abstract scenarios of use for walkthrough methods or to activate data flows or scenarios of location awareness. The distance between the researcher and the research persona can also afford a degree of cognitive shelter regarding the behavioral techniques of interface design. Generating and maintaining a research persona can, however, be challenging, as newly created personas may not trigger the same elements (especially ads) as more "organic" performances. The use of personas may also stand in contradiction to an apps' terms and conditions and can raise ethical concerns when interacting with other users. Moreover, while personas afford

some forms of distancing and enable conscious (re) staging of some aspects of the user experience, the researcher's status as user is clearly not exhausted by these stagings, nor is distancing guaranteed. Rather, researchers may be positioned as users in more or less conscious ways in different situations across entry points. Researchers must remain sensitive to these positionings or "configurations" (Woolgar, 1990) and learn when and how to best work with them in conducting research.

4. *Move from content to practices.* App research necessitates a renewed interest in the role of practices—particularly software-enabled mundane routines (Morris & Elkins, 2015)—as opposed to the study of content, which is prevalent in web and platform research. App interfaces enable specific, embodied, and often context-dependent activities. In this way, app stores sort and filter not necessarily content but practices. For example, a search for health apps in Google Play hardly returns informational apps; rather, the majority of health apps typically offer meditation, medical, self-tracking, and other tools and practice-focused solutions. While we recognize that there is no clear line between content and practice, apps tend to resemble dedicated equipment that support forms of *doing* as their privileged mode of engagement. Thus, when we study apps, we study the conditions of possibility for practices.

5. *Scale from situation to infrastructure and back.* The selected entry points allow the creation of research situations at multiple scales—from micro-practices enacted through interfaces to an aggregative bird's-eye view of app store analytics. While the app conceals itself as a bounded object, the app store brings it into relation with other objects by grouping the app with "Similar Apps," apps that "You May Also Like" or apps by the same developer. Individual apps can also belong to a larger group of apps (such as Facebook's "family of apps"), which are conceptualized as "app constellations" or collections of "mobile apps that share a single login and have app to app linking built in" (Wilson, 2014). There are also so-called "super apps," such as WeChat, which contain multiple services ranging from communication and finance to retail and social within a single app and require approaching such apps as sets of tightly connected modular services. Engaging with apps via multiple, connected situations may require navigating different scalar levels and relating specific findings in performative embodied situations (e.g., walkthroughs) to global app markets or cloud infrastructures. App research thus requires a heightened sensitivity toward scale and scope, toward where and how a particular method positions and is positioned.

6. *Engage with apps statically and dynamically.* Apps are run on devices that continuously collect data and make connections (through background sensing, updates, network-based calculations, and so forth); some of these result from specific practices and personas, while others do not. We differentiate between methods that can be considered "dynamic" (deployed within a "native" situation) and "static" (extracted from a native situation). Such a distinction is significant because dynamic app studies require a more deliberate effort in controlling the conditions of the situation by setting up personas, re-enacting physical locations through VPNs, or using specific research phones or software environments. In contrast, static approaches such as APK research or repurposing app (store) analytics do not require the enactment of native use situations.

7. *Resist presentism.* The fast update cycles of apps and the design decision of app stores to display only the latest versions creates a temporality persistently focused on the here and now. The stores' obfuscation techniques to prevent access to app packages and previous versions poses challenges to historical research. App repositories may function as access points to archives for downloading older versions, but they are often incomplete or contain potentially illegal and spammy software. Moreover, apps may come with limited rollouts and exist in different versions, while specific packages are highly dependent on externally loaded dynamic resources: emulating old app versions may load current assets in an old framework, mixing up some aspects of old apps with "current" material. In addition, there are temporal dimensions to the treatment of apps by app stores and their ranking algorithms, as an app's "freshness" may determine its ranking, and app rankings do change over time.

8. *Contest Silicon Valley imperialism.* The dominance of the Silicon Valley-native App Store and Google Play hides the fragmented and culturally and technically specific landscape of app stores globally. Each operating system, device manufacturer, country, and type of app may have its own app store that comes with specific infrastructural affordances and resistances for doing app research. For example, the alternative Russian app store for Android devices, Yandex.Store, does not offer a web interface to enable data collection, and China has banned Google Play. This has given rise to local tech giants such as Tencent's App Store, which requires specific language skills or registration processes on the researcher's end. The distinct regional infrastructural arrangements of app stores pose challenges but also opportunities to expand the scope of app studies. Future multi-situated approaches would benefit from going beyond

apps and infrastructures born out of Silicon Valley, which may require reworking some of the proposed methods and entry points.

9. *Don't leave ethics behind.* The proposed multi-situatedness confronts researchers with ethical problems and concerns across various domains, from the social and legal implications from handling large-scale network connection data to providing personal information for sensible registration processes. While there is no straightforward ethical checklist for multi-situated app research, our entry points and propositions raise various ethical considerations. For example, which ethical considerations arise when generating app data that are not "ready made" for social enquiry? When we encounter infrastructural resistance, are there ethical reasons to "leave things be"? Which ethical questions arise when researchers embody different roles (user, persona, researcher) and initiate different types of engagement with the objects of study? What ethical questions relate to the different practices that apps facilitate? The tech giants, moreover, have a poor record of operating "ahead" of regulatory environments; while operating in this regulatory limbo, researchers must take special care not to unwittingly replicate the ethical deficit that can be built into apps and their situations. We must precisely not "move fast and break things" even while following in the paths of those who do.

Conclusion

Ten years after the launch of Apple's App Store, the app economy is a billion-dollar global industry. Apps are so thoroughly insinuated into everyday life, they are often imperceptible: we seamlessly chat, take pictures, listen to music, play games, check our bank balance, and so on, with rarely a moment of reflection. Indeed, precisely because of their tendency to habituate use and background their operations, there is a critical need to re-situate apps to perceive how they work, generate value, and create the conditions of possibility for practice. We need to understand what is specific about their infrastructural embeddedness and how they operate within different sites and involve a diversity of often obscured stakeholders. In this article, we have suggested four entry points for researching apps through a general methodological framework that involves re-situating apps in ways that make them amenable for research.

There are, moreover, multiple opportunities to further expand this framework. For instance, integrated development environments as a key entry point on the developer side can be repurposed in any number of ways. Studies could explore entry points that require what might be described as "geo-situating"—a "dynamic" method that requires physically moving between locations (or emulating such movement) to explore geo-fencing, localization, and related dynamics. It

should be stressed that our approach here has also mainly addressed mobile apps, but it might offer inspiration for investigations into the increased embeddedness of apps across different software ecosystems, including the industrial and infrastructural settings associated with sensor-based media, smart cities, and the Internet of things. On a theoretical level, empirically informed app research offers opportunities to rethink investments around medium specificity and methodology. It does so by recognizing the diversity of data forms that converge in app usage as well as the unique engineered qualities of transformative digital objects such as packages with their infrastructural embeddedness that deliver a vast range of concretized relations and groupings. Despite the many differences between the entry points and methods covered, our approach is unified by a commitment to unpacking the infrastructural embeddedness of apps and with an eye on political economy. This can extend and diversify app studies, particularly in relation to the many interpretive studies of single apps. As such, additional multi-situated approaches to apps and infrastructures could further enrich our understanding of how apps are entangled with everyday practices.

ORCID iDs

Anne Helmond https://orcid.org/0000-0003-4327-4012
Fernando N. van der Vlist https://orcid.org/0000-0003-1401-0325

References

Akrich, M. (1992). The de-scription of technical objects. In W. E. Bijker & J. Law (Eds.), *Shaping technology/building society: Studies in sociotechnical change* (pp. 205–224). Cambridge, MA: MIT Press.

Allix, K., Bissyandé, T. F., Klein, J., & Le Traon, Y. (2016). AndroZoo: Collecting millions of Android apps for the

research community. In Proceedings of the 13th international conference on mining software repositories (pp. 468–471). New York, NY: ACM. doi:10.1145/2901739.2903508

Andersen, C. U., & Pold, S. B. (2018). *The metainterface: The art of platforms, cities, and clouds*. Cambridge, MA: MIT Press.

App Annie. (2018, January). 2017 Retrospective: A monumental year for the app economy. Retrieved from https://www.appannie.com/en/insights/market-data/app-annie-2017-retrospective/

App stores and their bias: Repurposing "app relatedness"? (2018). Retrieved from https://wiki.digitalmethods.net/Dmi/SummerSchool2018AppStoresBias

App stores as data infrastructure. (2017). Retrieved from https://wiki.digitalmethods.net/Dmi/WinterSchool2017AppStoresAsDataInfrastructure

Basole, R. C., & Karla, J. (2011). On the evolution of mobile platform ecosystem structure and strategy. *Business & Information Systems Engineering*, *3*, 313–322. doi:10.1007/s12599–011–0174–4

Baumer, E. P. S., Ames, M. G., Burrell, J., Brubaker, J. R., & Dourish, P. (2015). Why study technology non-use? *First Monday*, *20*. doi:10.5210/fm.v20i11.6310

Berners-Lee, T. (1996). WWW: Past, present, and future. *Computer*, *29*, 69–77. doi:10.1109/2.539724

Bodle, R. (2011). Regimes of sharing. *Information, Communication & Society*, *14*, 320–337. doi:10.1080/1369118X.2010.542825

Book, T., Pridgen, A., & Wallach, D. S. (2013). *Longitudinal analysis of Android ad library permissions* (Arxiv:1303.0857, 3030857). Retrieved from http://arxiv.org/abs/1303.0857

Bucher, T., & Helmond, A. (2018). The affordances of social media platforms. In J. Burgess, T. Poell, & A. Marwick (Eds.), *The SAGE handbook of social media* (pp. 233–253). London, England: SAGE.

Callon, M. (1984). Some elements of a sociology of translation: Domestication of the scallops and the fishermen of St Brieuc Bay. *The Sociological Review*, *32*, 196–233. doi:10.1111/j.1467-954X.1984.tb00113.x

Chao, J. (2018, December). Appcestry—A tool for the study of mobile application similarities. Retrieved from https://medium.com/jason-chaos-depository/appcestry-a-tool-for-the-study-of-mobile-application-similarities-a947bebda29e

Cooper, A. (2004). *The inmates are running the asylum: Why high tech products drive us crazy and how to restore the sanity*. Indianapolis, IN: Sams.

Dat[a]ing: Mapping data infrastructures of the dating industry. (2017). Retrieved from https://wiki.digitalmethods.net/Dmi/DataAndDating

Daubs, M. S., & Manzerolle, V. R. (2015). App-centric mobile media and commoditization: Implications for the future of the open Web. *Mobile Media & Communication*, *4*, 52–68. doi:10.1177/2050157915592657

Dieter, M. (2015). Dark patterns: Interface design, augmentation and crisis. In D. M. Berry & M. Dieter (Eds.), *Postdigital aesthetics* (pp. 163–178). Basingstoke, UK: Palgrave Macmillan.

Digital methods for app analysis: Mapping app ecologies in the Google Play Store. (2015). Retrieved from https://wiki.digitalmethods.net/Dmi/SummerSchool2015DigitalMethodsAppAnalysis

Enck, W., Gilbert, P., Han, S., Tendulkar, V., Chun, B.-G., Cox, L. P., & Sheth, A. N. (2014). TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones. *ACM Transactions on Computer Systems*, *32*(2), 1–29. doi:10.1145/2619091

Fagerjord, A. (2015). The cloud, the store, and millions of apps. In I. Kaldrack & M. Leeker (Eds.), *There is no software, there are just services* (pp. 91–102). Lüneburg, Germany: Meson Press.

Farman, J. (2015). Infrastructures of mobile social media. *Social Media + Society*, *1*(1). doi:10.1177/2056305115580343

Feuz, M., Fuller, M., & Stalder, F. (2011). Personal web searching in the age of semantic capitalism: Diagnosing the mechanisms of personalisation. *First Monday*, *16*(2). doi:10.5210/fm.v16i2.3344

Fuller, M. (Ed.). (2008). *Software studies: A lexicon*. Cambridge, MA: MIT Press.

Gehl, R. (2014). *Reverse engineering social media: Software, culture, and political economy in new media capitalism*. Philadelphia, PA: Temple University Press.

Gerlitz, C., Helmond, A., van der Vlist, F. N., & Weltevrede, E. (2016). App support ecologies: An empirical investigation of app-platform relations. In *1st 2016 annual conference of the collaborative research centre "media of cooperation."* Siegen, Germany: Artur-Woll-Haus, University of Siegen. Retrieved from https://bit.ly/app-support-ecologies

Gießmann, S., Röhl, T., & Trischler, R. (Eds.). (2019). *Materialität der Kooperation* [Materiality of cooperation]. New York, NY: Springer.

Goffman, E. (1981). *Forms of talk*. Philadelphia: University of Pennsylvania Press.

Hayles, N. K. (2017). *Unthought: The power of the cognitive nonconscious*. Chicago, IL: The University of Chicago Press.

Helmond, A. (2015). The platformization of the web: Making web data platform ready. *Social Media + Society*, *1*(2). doi:10.1177/2056305115603080

Helmond, A. (2017). Historical website ecology: Analyzing past states of the web using archived source code. In N. Brügger (Ed.), *Web 25: Histories from the first 25 years of the world wide web* (pp. 139–155). New York, NY: Peter Lang.

Hui, Y. (2016). *On the existence of digital objects*. Minneapolis: University of Minnesota Press.

Kagdi, H., Collard, M. L., & Maletic, J. I. (2007). A survey and taxonomy of approaches for mining software repositories in the context of software evolution. *Journal of Software Maintenance and Evolution: Research and Practice*, *19*, 77–131. doi:10.1002/smr.344

Karppi, T. (2018). *Disconnect: Facebook's affective bonds*. Minneapolis: University of Minnesota Press.

Light, B., Burgess, J., & Duguay, S. (2016). The walkthrough method: An approach to the study of apps. *New Media & Society*, *20*, 881–900. doi:10.1177/1461444816675438

Ma, Z., Wang, H., Guo, Y., & Chen, X. (2016). LibRadar: Fast and accurate detection of third-party libraries in Android apps. In *Proceedings of the 38th 2016 international conference on software engineering companion* (pp. 653–656). New York, NY: ACM. doi:10.1145/2889160.2889178

Mackenzie, A. (2010). *Wirelessness: Radical empiricism in network cultures*. Cambridge, MA: MIT Press.

Mapping (secure) messaging app ecologies. (2016). Retrieved from https://wiki.digitalmethods.net/Dmi/SummerSchool2016SecureMessagingAppEcologies

Mapping data-intensive app infrastructures. (2018). Retrieved from https://wiki.digitalmethods.net/Dmi/WinterSchool2018MappingDataIntensiveAppInfrastructures

Marcus, G. E. (1995). Ethnography in/of the world system: The emergence of multi-sited ethnography. *Annual Review of Anthropology*, *24*, 95–117.

Marres, N. (2015). Why map issues? On controversy analysis as a digital method. *Science, Technology, & Human Values*, *40*, 655–686. doi:10.1177/0162243915574602

Matviyenko, S., Ticineto Clough, P., & Galloway, A. R. (2015). On governance, blackboxing, measure, body, affect and apps: A conversation with Patricia Ticineto Clough and Alexander R. Galloway. *The Fibreculture Journal*, *25*, 10–29. doi:10.15307/fcj.25.179.2015

Miller, P. D., & Matviyenko, S. (Eds.). (2014). *The imaginary app*. Cambridge, MA: MIT Press.

Montfort, N., Baudoin, P., Bell, J., Bogost, I., Douglass, J., Marino, M. C., & Vawter, N. (2012). *10 PRINT CHR$, 205.5+RND, 1)); : GOTO 10*. Cambridge, MA: MIT Press.

Morris, J. W., & Elkins, E. (2015). There's a history for that: Apps and mundane software as commodity. *The Fibreculture Journal*, *25*, 63–88. doi:10.15307/fcj.25.181.2015

Morris, J. W., & Murray, S. (Eds.). (2018). *Appified: Culture in the age of apps*. Ann Arbor: University of Michigan Press.

Nicolini, D. (2017). Is small the only beautiful? Making sense of "large phenomena" from a practice-based perspective. In A. Hui, T. Schatzki, & E. Shove (Eds.), *The nexus of practices* (pp. 98–113). New York, NY: Routledge.

Nieborg, D. B. (2015). Crushing candy: The free-to-play game in its connective commodity form. *Social Media + Society*, *1*(2). doi:10.1177/2056305115621932

Nodder, C. (2013). *Evil by design: Interaction design to lead us into temptation*. Indianapolis, IN: John Wiley.

Parks, L., & Starosielski, N. (Eds.). (2015). *Signal traffic: Critical studies of media infrastructures*. Champaign: University of Illinois Press.

Rieder, B., Matamoros-Fernández, A., & Coromina, O. (2018). From ranking algorithms to "ranking cultures": Investigating the modulation of visibility in YouTube search results. *Convergence*, *24*, 50–68. doi:10.1177/1354856517736982

Rochet, J.-C., & Tirole, J. (2003). Platform competition in two-sided markets. *Journal of the European Economic Association*, *1*, 990–1029. doi:10.1162/154247603322493212

Rogers, R. (2013). *Digital methods*. Cambridge, MA: MIT Press.

Suchman, L. (2007). *Human-machine reconfigurations: Plans and situated actions*. Cambridge, UK: Cambridge University Press.

The tracker's guide to the cloud. (2014). Retrieved from https://digitalmethods.net/Dmi/TrackersGuide

Vannini, P. (2008). Situatedness. In L. M. Given (Ed.), *The SAGE encyclopedia of qualitative research methods* (pp. 815–816). Thousand Oaks, CA: SAGE. doi:10.4135/9781412963909

Venturini, T., Jacomy, M., Meunier, A., & Latour, B. (2017). An unexpected journey: A few lessons from Sciences Po médialab's experience. *Big Data & Society*, *4*(2). doi:10.1177/2053951717720949

Vonderau, P. (2018). App Annie: App industry rankings and infrastructure. In J. W. Morris, J. W., & S. Murray (Eds.), *Appified: Culture in the age of apps* (pp. 23–31). Ann Arbor: University of Michigan Press.

Weltevrede, E. (2016). *Repurposing digital methods: The research affordances of platforms and engines* (Doctoral dissertation, University of Amsterdam, The Netherlands). Retrieved from https://wiki.digitalmethods.net/Dmi/RepurposingDigitalMethods

Wilson, F. (2014). *App constellations*. Retrieved from https://avc.com/2014/05/app-constellations/

Woolgar, S. (1990). Configuring the user: The case of usability trials. *The Sociological Review*, *38*(Suppl. 1), 58–99. doi:10.1111/j.1467-954X.1990.tb03349.x

Yeung, K. (2017). "Hypernudge": Big data as a mode of regulation by design. *Information, Communication & Society*, *20*, 118–136. doi:10.1080/1369118X.2016.1186713

Zittrain, J. (2009). *The future of the internet: And how to stop it*. New Haven, CT: Yale University Press.

Author Biographies

Michael Dieter (PhD University of Melbourne) is an assistant professor at the Centre for Interdisciplinary Methodologies (CIM), University of Warwick. His research interests include publishing practices after digitization, cultural techniques in interface and user-experience design, and genealogies of media at the intersection of aesthetic and political thought.

Carolin Gerlitz (PhD Goldsmiths, University of London) is a professor of Digital Media and Methods at the University of Siegen. Her research interests include digital media technologies, quantification, inventive methods, software studies, platforms, sensor media, and critical data studies.

Anne Helmond (PhD University of Amsterdam) is an assistant professor of New Media and Digital Culture at the University of Amsterdam. Her research interests include software studies, platform studies, app studies, digital methods, and web history.

Nathaniel Tkacz (PhD University of Melbourne) is a Reader of Digital Media and Culture at the University of Warwick. His research interests include apps, interface methods and criticism, platform economies, experience design, and critical data studies.

Fernando N. van der Vlist (MA University of Amsterdam) is a research associate with the Collaborative Research Centre "Media of Cooperation" at the University of Siegen. His research interests include software studies, digital methods, social media and platform studies, app studies, and critical data studies.

Esther Weltevrede (PhD University of Amsterdam) is an assistant professor of New Media and Digital Culture at the University of Amsterdam. Her research interests include digital methods, software and platform studies, app studies, data infrastructures, and social media automation.