

Working Paper Series
Collaborative Research Center 1187 Media of Cooperation

Print-ISSN 2567-2509

Online-ISSN 2567-2517

URN urn:nbn:de:hbz:467-13410



This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

This Working Paper Series is edited by the Collaborative Research Center Media of Cooperation and serves as a platform to circulate work in progress or preprints in order to encourage the exchange of ideas. Please contact the authors if you have any questions or comments. Copyright remains with the authors.

The Working Papers are accessible via the website <http://wp-series.mediacoop.uni-siegen.de> or can be ordered in print by sending an email to workingpaperseries@sfb1187.uni-siegen.de

Publication of the series is funded by the German Research Foundation (DFG).

The cover image is a modification of fig. 4 in this paper ›Companies behind the network connections coming in or out of four popular dating apps (OkCupid, Grindr, Tinder, BeeTalk)‹, designed and re-arranged by the authors.

Universität Siegen
SFB 1187 Medien der Kooperation
Herrengarten 3
57072 Siegen, Germany
www.sfb1187.uni-siegen.de
workingpaperseries@sfb1187.uni-siegen.de

Store, interface, package, connection

Methods and propositions for multi-situated app studies

Michael Dieter	 0000-0002-5894-7892	 @mdieter
Carolin Gerlitz	0000-0002-5716-9355	@cgrltz
Anne Helmond	0000-0003-4327-4012	@silvertje
Nathaniel Tkacz	0000-0002-5796-4105	@__nate__
Fernando van der Vlist	0000-0003-1401-0325	@fvandervlist
Esther Weltevrede	0000-0001-5276-297X	@esthr

Abstract This paper discusses methodological approaches to app studies, focussing on their embeddedness and situatedness within multiple infrastructural settings. Our approach arises by paying 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, makes apps visible and accountable in a variety of unique ways. Engaging with and even staging these situations, therefore, allows for political-economic, social and cultural dynamics associated with apps and their infrastructures can be investigated through a style of research we describe as multi-situated app studies. The piece offers an overview of four different entry points of enquiry that are exemplary of this overarching framework, focussing 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

July 10th, 2018 marked the ten-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 after, Google launched Android Market on October 22, 2008 – which was later rebranded as Google Play – and similarly made apps for the Android operating system available. 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 infor-

mational *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, containing over 3.8 million Android apps and 2 million iOS apps that generate over 86 billion USD in revenue. With the average user spending almost 1.5 months using apps per year, 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). Most importantly, apps pose several empirical challenges for media

research because of this tendency to move into the background, while still remaining thoroughly entangled with data-intensive infrastructures and new economic models like platforms.

Apps are designed to perform as concrete software objects, yet are continually transformed through interaction 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, through the organised 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 individually activate them by pressing their bounded 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 milieu 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 that 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 woven throughout the entire object, but one that also supports a certain kind of multivalence as it enables its stakeholders including app stores, developers, partners or users, among others, to integrate and valorise 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 concretised systems of relations in standardised 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 immediately exist as digital objects within technical milieu, it is less a case of following an app, than re-situating it drawing from a number of unique affordances are available to the researcher. Indeed, the kinds of reciprocal causality initiated by apps is

in general highly controllable due to its 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 and visibility. Even while encountering at times forms of resistance and constraints (as we will go on to discuss), it is possible as a result to tease out different forms, values and relations that apps can take by staging experimental and exploratory situations.

In this paper we call this a multi-situated approach to apps, where each methodological orientation or ‘entry point’ at the same time *deploys* and *makes visible* different infrastructural settings. Situatedness, in our understanding, includes the common understanding as ‘the involvement of the researcher within a research site’ (Vannini, 2008, p. 815), but also indicates the standardized, yet differentiated technical agencies at work when using apps. In what follows we present four different methodological entry points through which researchers can actively invoke different app situations and use these to advance or initiate an inquiry. This may entail performative usage of an app through purposefully orchestrated, personalised user experiences, but might also using data flows originally designed for machine reading. The selection of entry points draws on key sites within which many stakeholders engage with apps, yet 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 paper summarizes and 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 are in stores like 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. App stores, in this way, function as key gatekeepers – or indeed, as ‘obligatory passage points’ (Callon, 1984; Fagerjord, 2015) – by setting up the rules for app creation, sorting, and distribution, and do so by drawing from the economic model of the multi-sided marketplace (Rochet & Tirole, 2006) to ensure app exposure to potential customers. In order to account for the research affordances (Welterde, 2016) of app stores, we need to understand the specific situations that app stores create for apps. Indeed, they allow for a multi-dimensional perspec-

tive on the relations between apps as organised by the assemblage of stores, their algorithms, rankings, developer and user activities. App stores can, accordingly, be used for addressing relations among apps as issue spaces, for 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 paper, 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.¹ While app stores are often considered as 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 insights companies like 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. 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., App Store), or only have device-based mobile interfaces (e.g., Yandex.Store). Additionally, most app stores do not offer systematic access to their data via standard APIs, with Apple's App Store being one exception.² This means 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: for instance, by searching for specific apps (e.g., [Facebook]), topics (e.g., [pregnancy]), genres (e.g., [messenger]), 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, introducing research affordances for exploring app collections based on the demarcation of the store, and following possible user pathways. Users are also presented with additional app groupings on individual app pages. For example, Google Play has personalised 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 a section 'More By This Developer' and 'You May Also Like' (technically specified as '/customers-also-bought'). A key challenge for working with app relatedness are the personalisation and localisation effects of app stores – similar to search engine research – which can determine their results based on location, country, language, but also previous user behaviour. Central to, but also beyond, the algorithmic ordering of apps are the categories to which developers can assign their apps, which then inform similarity calculations, but also the users' engagement with apps by browsing these categories. With these various sorting and ordering mechanisms, apps can be understood in not just a single bounded form, but as a digital object that can be diversely situated, connected to and related with through different queries, topics, categories, developer categorisation, as well as user behaviour.

App stores enable researchers to draw on, but also 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 the 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. Using the DMI Google Play Similar Apps and iTunes Store tools,^{3,4} we are able to extract the details of individual apps, collect 'Similar Apps', and extract their details as well. This similar app data can then be used to identify networks of app relations as created by the store.

¹ <https://wiki.digitalmethods.net/Dmi/WinterSchool2017AppStoresAsDataInfrastructure>

² <https://affiliate.itunes.apple.com/resources/documentation/itunes-store-web-service-search-api/>

³ <https://wiki.digitalmethods.net/Dmi/ToolGooglePlaySimilar>

⁴ <https://wiki.digitalmethods.net/Dmi/ToolItunesStore>

Queries can be deployed to engage with collections of apps as opposed to focussing on single apps in isolation. First, stores can be used to generate thematic and issue-oriented collections which can be studied as expressions or indicators of cultural difference. For example, how are apps employed in the performance of religion (e. g., Buddhism, Christianity, Hinduism, Islam, and Judaism)?⁵ What solutions do app stores – as indices of apps – generate or recommend when querying for various controversial, partisan, objectionable content, or sensitive issues (e. g., abortion, gun control, porn, or mental disorders and conditions)?⁶ The latter, in particular, can be understood as an application of issue mapping (Marres, 2015) from within the organisational 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 and more detailed affordances to engage with the question what features apps offer, as they usually provide insights into their key functionality and interoperability with other apps, platforms or infrastructures. How do different secure messaging apps position themselves in their self-descriptions and to what extent – and in what terms – do they address issues around security, encryption, and usability?⁷ App descriptions can be used as starting point for manual categorisation, for topic modelling through natural language processing or be queried for predefined terms and phrases.

Third, a large proportion of apps are not created as standalone objects (only), but are built on top of, or in relation, to other apps, software or platforms, for instance by drawing on the application programming interfaces (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 top of platforms and intensify, support, interpret, alter, or amend their features, data, and associated practices (Gerlitz et al., 2016).

But the sorting processes of apps can also be made subject to empirical enquiry themselves, opening up questions around how ‘app relatedness’ is produced in the first place and how the algorithmic politics compare across the different app stores or change over time. Although algorithmic processes are generally difficult to account for and interpret, it is pos-

sible to repeatedly collect or log search results and their rankings as they unfold over time (cf., Rogers, 2013). 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 into the curation and removal of certain apps by app stores.⁸ Moreover, such approaches do not only address ranking algorithms, but also ‘ranking cultures’ – highlighting the ‘distributed and heterogeneous agencies that converge’ in ranked lists (Rieder et al., 2018).

The research affordances for collection making and indexing apps through app stores opens possibilities to generate research situations that provide insights into the relational and infrastructural situatedness of apps. As outlined above, app stores allow for the identification of how topics, issues or genres are addressed through apps, or what app developers offer as packaged solutions, which practices apps support and how app stores relate these apps to one another. Interestingly, engaging with apps through app stores resurfaces the role of practices as app store algorithms not only sort and organise content, but also modes of engagement and user practices (with regards to topical concerns, with regards to social media) as apps are designed to structure behaviour and not meaning.

App interfaces

While app stores support research into the relations between apps and their economisation, app interfaces offer entry points for specific inquiries into the conditions of possibilities for user practices. It has been a longstanding 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 that are especially pertinent under conditions of platformisation and the data-intensive forms of ‘controlled consumption’ that apps facilitate (Andersen & Pold, 2018). The walkthrough method assists with exploring these questions by systematically documenting and abstracting interface features in their normative infrastructural settings.

The walkthrough method is commonly used in software engineering and user-centred design research to present a software product to peers or

⁵ <https://wiki.digitalmethods.net/Dmi/SummerSchool2015DigitalMethodsAppAnalysis>

⁶ <https://wiki.digitalmethods.net/Dmi/SummerSchool2018AppStoresBiasObjectionableQueries>

⁷ <https://wiki.digitalmethods.net/Dmi/SummerSchool2016SecureMessagingAppEcologies>

⁸ <https://wiki.digitalmethods.net/Dmi/SummerSchool2018AppStoresBias>

stakeholders for review. It is also used in commercial technology reviews and has a longer history in product demonstrations and infomercials. Ben Light et al. (2016) have suggested, however, that walkthroughs can be repurposed in a 'significant departure' from these prior uses in order 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 contextualised 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 studies 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 behavioural design regimes as a mode of scripting the user that targets the nonconscious dimensions of cognition (Hayles, 2017), leveraging insights from behavioural economics, cognitive psychology, and neurological research, while operationally relying on big data and nudging (Yeung, 2017). In this regime of design, 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, behavioural 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 recognise 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 empirical research into a market or audiences for a product, then imagining an ideal type that can be utilised to develop this 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 personalisation – but also 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 which 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.

While 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, as indicated in the examples provided thus far, we see several opportunities arising from moving from single to *comparative* app 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 organised (action points, input buttons, notifications), how design patterns are implemented, or how navigation paths are arranged. To illustrate, Figure 1 (p. 6) 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 personas – user, developer, advertiser, and so on – in order to consider how platform providers address their varying groups on different sides via distinct interfaces (cf., Bucher & Helmond, 2018). By de-emphasising the user-centred 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 inquiry, such as how apps reconfigure and regulate platform labour 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 like 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 allow for data input (Gehl, 2014), including touchpoints, buttons and forms that might be emphasized to indicate platform relations like the presence of Facebook or Google logins that speak to the techno-economic relations of platformisation (Helmond, 2015). As an example, Figure 2 (p. 7) is a series of comparative walkthroughs of dating apps that explore how they are situated within specific data infrastructures.⁹ In this case, the main concern involves tracing the magnitude and pacing of inbound and outbound data flows, linking the micro-practice

⁹ <https://wiki.digitalmethods.net/Dmi/DataAndDating>



Fig. 1: Side-by-side comparison of login or sign-up screens in a selection of mindfulness apps.

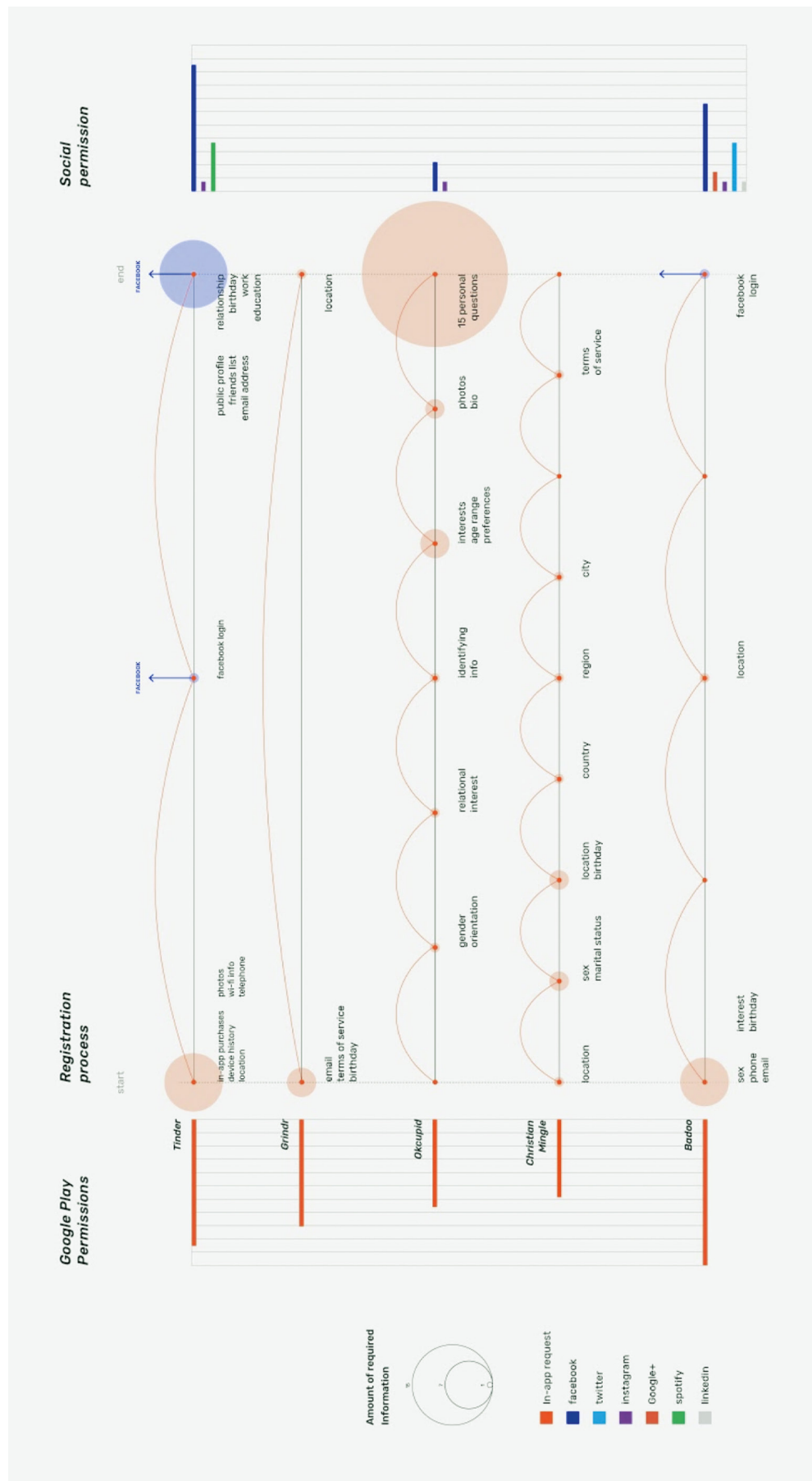


Fig. 2: Comparative information visualisation of the user registration process in dating apps (Tinder, Grindr, OkCupid, Christian Mingle, Badoo).

of interface engagement with the distribution of personal information. The visualisation strategy, accordingly, abandons the GUI-screen capture in favour of a more data-centric approach, enabling the quantification and comparisons of informational disclosure patterns across a set of related apps.

Emphasizing the situatedness of app interfaces can, accordingly, be further developed through experimental visualisations 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 utilised, for instance, to address particular challenges in the analysis of user journeys like 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, annotate or ‘markup’ the walkthrough are, we suggest, not to be reduced to images and analysed with semiotic methods. Apps are first and foremost operational media; they are *applications*, things for doing. Importantly, apps are designed with behaviours – not meanings – in mind. App developers aim to get their users to do specific things – to change their behaviour – and the walkthrough method can be used to reflect this behavioural focus. While user experience, usability, and cultural studies-inspired approaches place the user at the centre of the research, we see potential in using walkthroughs to examine how apps are infrastructurally situated by teasing out data flows, design strategies and platform logics 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.

App packages

Our third entry point allows for the exploration in more detail 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, but rather 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 utilising 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 Kit) 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 the app stores where they can be subjected to review before being admitted to these 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 using 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 utilising a third-party app manager such as Cydia (iOS) or Cydia Impactor (desktop). Android, on the other hand, positions itself as an open platform and developers can distribute their apps via various third-party Android app stores and marketplaces, or via their own websites.¹⁰ Users can then configure their device settings to download and install ‘unknown’ apps from 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 the 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 et al., 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 apps, but only offer a few versions of an app, while APKMirror appears narrower in scope but with many versions of the most popular apps. This is to suggest that each repository is to be considered on its own terms; none operate as perfect archives and each has different affordances for doing (historical) app research. Within software engineering, software repositories are an important source for studying the

¹⁰ <https://developer.android.com/distribute/marketing-tools/alternative-distribution>

evolution of software (Kagdi et al., 2007). This immediately raises issues about running or emulating older APK packages to examine the larger techno-commercial ecosystems apps are embedded in over time or analyse 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 or not. 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 authenticating with a Google Account to retrieve apps and will only provide 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 like 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 in order to fully view their contents and this can usually be achieved with the support of additional tools.¹¹ Decoding apps shows the contents of the package including all the necessary files and resources. The `AndroidManifest.xml` file, for example, describes metadata like the name, version, and contents of the APK file and also includes information about the app's permissions. Further resources in the package include software development kits (SDKs) used to build particular modules in the app, including social logins, app analytics and advertisement libraries.

Re-situating apps as software makes them available to the range of enquiries found in critical code and software studies (Montfort et. al., 2012; Fuller, 2008). One can study the code up close, or parse it through other diagnostic tools enabling comparisons across apps or sets of apps (e.g., Appcestry). 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 present purposes, app packages can be used to examine the various stakeholders involved in the production of apps (multi-sidedness) as well as the infrastructures for transferring and storing app content and data (multi-sidedness). In particular, through analysing inscribed libraries like 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 optimisation 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 its list of known tracking technologies and generates reports for individual apps. Another tracker tool, the DMI App Tracker Tracker, is built on top of 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 a lot of related work in computer science and software engineering examining third-party libraries, including advertising libraries, in apps (Book et al., 2013; Ma et al., 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.¹² Additionally, 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, business model pivots, or otherwise reflect dynamics in the wider economies of app advertising, app development, app analytics, and mobile game monetisation.

App connections

Our fourth entry point for multi-situated app studies are the network connections that mobile devices establish and that allow both multi-sidedness and multi-sidedness 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

¹¹ <https://ibotpeaches.github.io/Apktool/>

¹² <https://wiki.digitalmethods.net/Dmi/WinterSchool2018MappingDataIntensiveAppInfrastructures>

each and every access point that is listening for such access requests. As Mackenzie points out, the implication is that users' devices are continuously connecting and disconnecting to objects and infrastructures 'without knowing exactly how or where' (2010, p. 5). When approaching apps as being multi-situated in this sense, network connections are a key entry point for understanding how apps are always and necessarily bound up with – or indeed '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. In order 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, analyse, 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 like 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.

Previous research on tracking technologies, cloud infrastructures, and data infrastructures is based mainly on web corpora.¹³ However, some of these approaches may be resensitised to explore the multiple sites and sides of 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 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 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 personalised 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 permissions required by most operating systems to establish network connections are approved during installation. Interestingly however, internet permissions are classified as 'normal' on Android as one of the standard permissions – that is, 'permissions that don't pose much risk to the user's privacy or the device's operation' (Android Developers) – that apps need and are granted automatically (and cannot be opted out!).¹⁴ Additionally, apps might not require all the permissions they request and might access more data than is functionally needed.

The boundedness of apps as bundles or packages is challenged when recognising that apps are routinely extending themselves through these network connections. On the 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 characterise 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. Additionally, there has been a growing interest in studying the materiality of internet infrastructure and signal traffic (Parks & Starosielski, 2015). What or who do these connections serve? Furthermore, researchers can also inspect or even intercept the data packets transmitted unsecurely across these network connections with packet inspection methods. Using common network data packet inspection tools like Wireshark and tcpdump, we collected and analysed query parameters in HTTP requests, which among other things yielded detailed ad requests to

¹³ <https://digitalmethods.net/Dmi/TrackersGuide>

¹⁴ <https://developer.android.com/reference/android/Manifest.permission#INTERNET>

```
http://ads.mopub.com/m/ad?v=6&id=7d8c0acc4e3248119c29
94578999a413&nv=4.11.0&dn=LGE%2CNexus%205%2CCha
mmerhead&bundle=com.grindrapp.android&q=m_gender%3
Am%2Cm_age%3A34&ll=52.3690466%2C4.8934122&lla=19
&llf=450836&llsdk=1&z=%2B0200&o=p&w=1080&h=1920&s
c_a=3.0&mcc=204&mnc=16&iso=nl&cn=T-Mobile%20%20NL
&ct=2&av=3.10.0&udid=ifa%3Abf58ff79-eb26-4e26-bb81-3ffe
f7ba2154&dnt=0&mr=1&android_perms_ext_storage=1
```

Fig. 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).

ad networks (Figure 3).¹⁵ Such data is accessible through any unencrypted HTTP connection and may easily be captured.

Some network utility apps like Network Connections¹⁶ (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. But one could also conceive of rich and mature profiles trained to enact a researcher’s choreographed situation (e.g., to trigger cookies, personalisation, localisation, targeted ads). Some of these strategies were originally developed for studying personalisation and localisation in search engine results (Feuz, Fuller, & Stalder, 2011; Rogers, 2013). Once network connections data are obtained through dedicated tools, it is sometimes also possible to trace them back to the firms and organisations behind them (Figure 4, p. 12).¹⁷ 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 centres, cloud servers) that are being connected to? And which buttons – or what kind of scripts – trigger these infra-

structural relations for serving content, serving ads, signing in, and sharing content?

Towards situated app studies: nine propositions

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 towards 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 data points ready-made for ‘social’ investigations and accessible in structured ways via open APIs, apps are rather characterised 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 social data, but mostly do not offer structured access via open APIs, which means there is an absence of data ready-made for both developers and researchers. Apps that present social data through their interfaces (e.g. dating apps, chat and instant messaging apps) often cannot be easily retrieved as the alternative 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 as being 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 present researchers with the challenge to devise inventive methods that respond to the heterogeneity, to the different kinds of structure, and multiple origins of data in which ready-made data are both rare and in need of unpacking.

(2) *Navigate infrastructural resistance.* While obfuscation is a commonplace technique in computer science and software engineering (Matviyenko et al., 2015), the deliberate efforts or infrastructural effects that render code and data illegible to both human and technical interpretation have a significant impact on app research. Some forms of obfuscation are intentional: APK files might be designed to frustrate decompiling tools; certificate pinning and encrypted channels can limit the scope of packet sniffing; while entire app ecosystems like 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

¹⁵ <https://wiki.digitalmethods.net/Dmi/DataAndDating>; <https://www.digitalmethods.net/Dmi/WinterSchool2018MappingDataIntensiveAppInfrastructures>

¹⁶ <https://play.google.com/store/apps/details?id=com.antispycell.connmonitor>

¹⁷ <https://wiki.digitalmethods.net/Dmi/DataAndDating>

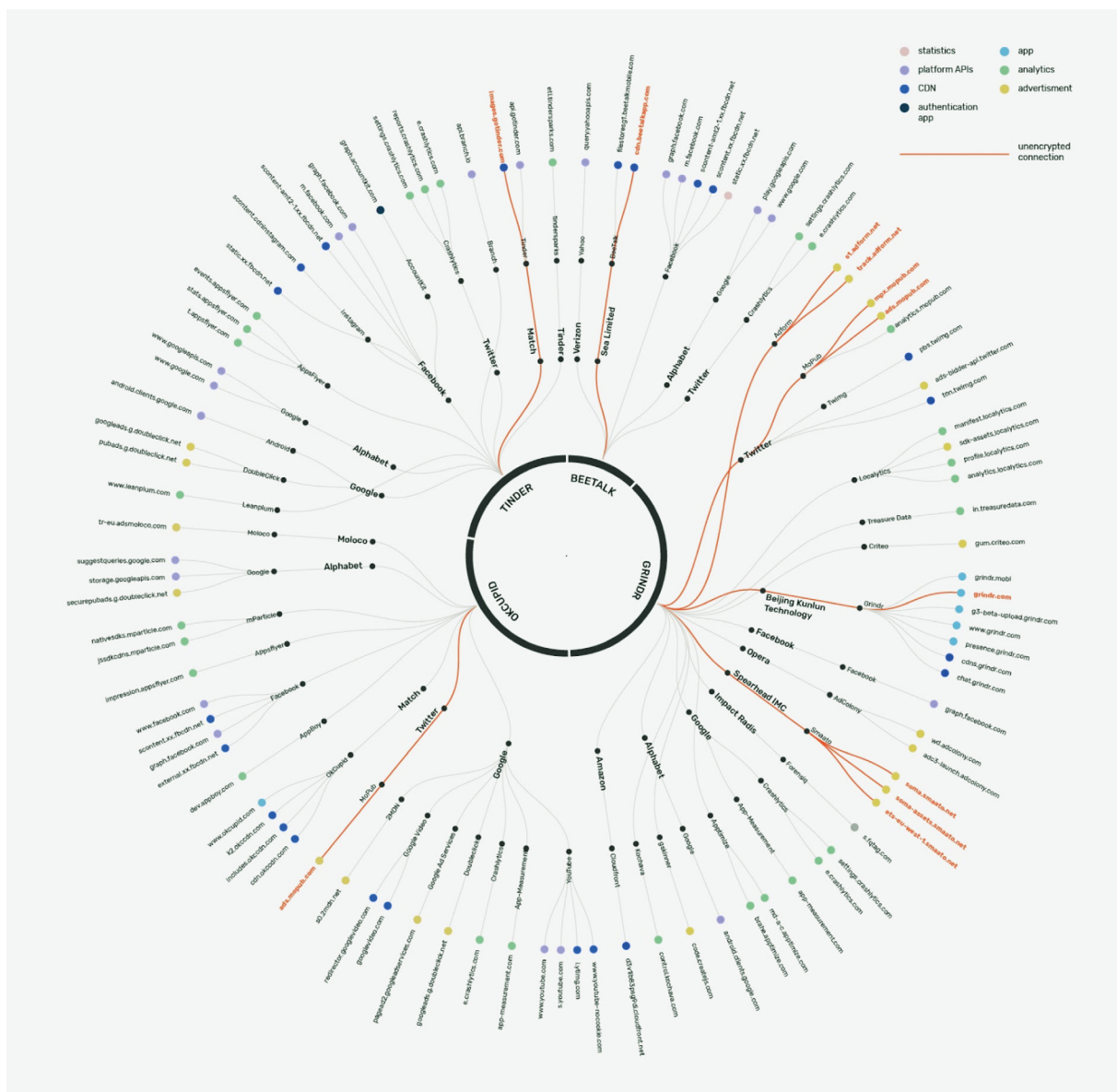


Fig. 4: Companies behind the network connections coming in or out of four popular dating apps (OkCupid, Grindr, Tinder, BeeTalk).

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 realise ANT's principle to 'follow the actors' (Venturini et al., 2017), apps pose various challenges to the idea of following actors (or things). But 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 resituate apps to open up alternative (albeit partial) perspectives. Producing situations, accordingly, may involve working both with and against the objectives and infrastructural resistance of apps.

(3) *(Un)do the user*. Empirical engagement with apps opens up a complicated relation to the user: while both software and platform studies have bracketed out users and their practices, app research requires a partial return of the user. Many apps require personalised logins, build on existing social media profiles, initiate data flows only through user practices, or are tied to users' specific locations. While disentanglement from user practices is possible in some situations (such as minimising the personalisation effects of app stores), or might simply not be an issue in others (APK-based methods that work without profiles), it is simply not possible when performing walkthroughs, studying network connections or sensor-based data flows, as all these build on situated practices. Apps present researchers with the paradox

that in order to systematically move away from user-centred research, they need to deliberately create 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 their research persona can also afford a degree of cognitive shelter regarding the behavioural 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.

(4) *Move from content to practices.* App research affords a renewed interest in the role of practices – particularly software-enabled mundane routines (Morris & Elkins, 2015) – as opposed to the study of content prevalent in web and platform research. App interfaces enable specific, embodied, and often context-dependent activities, and, in this way, app stores do not necessarily sort and filter content, but rather practices. For example, a search for health apps in Google Play hardly returns informational apps, but rather the majority of health apps typically offer meditation, medical, self-tracking and other tools and practice-focused solutions. While recognising 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’ – or 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’, conceptualised as ‘app constellations’ which are collections of ‘mobile apps that share a single login and have app to app linking built in’ (Wilson, 2014). Or so-called ‘super apps’ like WeChat, which contain multiple services ranging from communication and finance to retail and social

within a single app. Engaging with apps via multiple situations may require navigating different scalar levels and relating specific findings in performative embodied situations (such as walkthroughs) to global app markets or cloud infrastructures. App research thus requires a heightened sensitivity towards scale and scope, towards where and how a particular method 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 which are enabled by specific practices and personas, while others are 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. Static approaches such as APK research or repurposing app (store) analytics by contrast 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 only display 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 approaches. App repositories may function as access points to archives for downloading older versions, however 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 present material. In addition, there are temporal dimensions to the treatment of apps by app stores and their ranking algorithms, as app ‘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. 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 collec-

tion and China has banned Google Play, giving rise to local tech giants such as Tencent's App Store that 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 beyond the Bay Area.

(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. The methods outlined above cut across a number of disciplines and domains of practice, from computer science and media studies, to political economy and design, all of which have their own ethical norms and frameworks. When enacting multiple research situations, researchers need to reflect on how these frameworks might overlap or conflict, and how they might be effectively realigned according to specific research situations. 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 replicate the ethical deficit that can be built into apps and their situations. We must not 'move fast and break things'.

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, without any moment of reflection. Indeed, precisely because of their tendency to habituate use and background their operations, there is a critical need to resituate apps in order to perceive how they work, generate value, and create 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 resituating 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. Entry points that require what might be described as 'geo-situating' could be explored – a 'dynamic' method that requires physically moving between locations (or emulating such

movement) to explore geo-fencing, localisation, and related dynamics. It should be stressed that our approach here has also mainly addressed mobile apps, yet 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 recognising the diversity of data forms that converge in app usage as well as the unique engineered qualities of transformative digital objects like 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 or relations of apps. A better appreciation of this unique entanglements of apps and infrastructures remains an unfinished project, as does a more nuanced understanding of how these modulate the practices of everyday life. In this regard, which other situations are still in need of invention?

Acknowledgements

We would like to thank Emile den Tex and Erik Borra (Digital Methods Initiative, University of Amsterdam), and Jason Chao and James Tripp (Centre of Interdisciplinary Methodologies, University of Warwick) for their tool development, technical expertise, and commentaries. We would also like to thank all participants and designers in the referenced projects conducted during previous Digital Methods Summer and Winter Schools (2015–2018).

Funding

Parts of this work are supported by the Collaborative Research Centre 'Media of Cooperation' (SFB 1187), funded by the German Research Foundation (DFG) and the research programme Innovational Research Incentives Scheme Veni with project number 275-45-009, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

Declaration of conflicting interests

The authors declare that there is no conflict of interest.

Author biographies

Michael Dieter is assistant professor at the Centre for Interdisciplinary Methodologies (CIM) at the University of Warwick and holds a PhD in Culture and Communication from the University of Melbourne on media art criticism. With David Berry, he is the co-editor of the collection *Postdigital Aesthetics: Art, Computation and Design*, and his work has been published in the academic journals such as *differences*, *Fibreculture* and *Theory, Culture & Society*.

Carolyn Gerlitz is professor of Digital Media and Methods at the University of Siegen, Germany, and member of the Digital Methods Initiative Amsterdam researching platforms, quantification, sensor media, infrastructures, data critique, and inventive digital methodologies. Carolyn holds a NWO funded Veni grant for the project 'Numbering Life. Measures and Metrics in Digital Media'. She is PI and deputy speaker at the DFG funded collaborative research centre 'Media of Cooperation' (SFB 1187) and co-speaker of the graduate school 'Locating Media' (GRK 1769).

Anne Helmond is assistant professor of New Media and Digital Culture at the University of Amsterdam. She is a member of the Digital Methods Initiative research collective where she focuses her research on social media platforms, apps, and web history. She holds a Veni grant from the Netherlands Organisation for Scientific Research (NWO) for the project 'App ecosystems: A critical history of apps'.

Nathaniel Tkacz is reader at the Centre for Interdisciplinary Methodologies (CIM) at the University of Warwick. His work investigates the political, economic and organisational dimensions of technology, with a specific focus on networked and digital forms, the rise of 'dashboard interfaces' and the relationship between media and economy. He has published the book *Wikipedia and the Politics of Openness* (2014) with University of Chicago Press and has received an ESRC grant for the project 'Interrogating the Dashboard: Data, Indicators and Decision-making'.

Fernando van der Vlist is a doctoral researcher and research associate with the Collaborative Research Centre 'Media of Cooperation' (University of Siegen). He is a member of the Digital Methods Initiative (University of Amsterdam) and cofounder of the App Studies Initiative. He is a former lecturer in New Media and Digital Culture (University of Amsterdam, 2014–2018) and in Digital Methods (Utrecht University). His research is published in *Big Data & Society*, *Surveillance & Society*, *Global Networks*, and *Internet Policy Review*.

Esther Weltevrede is assistant professor of New Media and Digital Culture at the Department of Media Studies at the University of Amsterdam. Esther is also coordinating the Digital Methods Initiative (DMI), the new media research program at the Amsterdam School for Cultural Analysis (ASCA). Her re-

search interests include digital methods, app studies, data infrastructures and social media automation.

References

- Akrich, M. (1992). The De-Description of Technical Objects. In W. E. Bijker & J. Law (Eds.), *Shaping Technology/Building Society: Studies in Sociotechnical Change* (pp. 205–224). Cambridge, MA: The 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: The MIT Press.
- App Annie. (2018, January 17). 2017 Retrospective: A Monumental Year for the App Economy. Retrieved from <https://www.appannie.com/en/insights/market-data/app-annie-2017-retrospective/>
- Basole, R. C., & Karla, J. (2011). On the Evolution of Mobile Platform Ecosystem Structure and Strategy. *Business & Information Systems Engineering*, 3(5), 313. doi: 10.1007/s12599-011-0174-4
- Berners-Lee, T. (1996). WWW: past, present, and future. *Computer*, 29(10), 69–77. doi: 10.1109/2.539724
- Bodle, R. (2011). Regimes of Sharing. *Information, Communication & Society*, 14(3), 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 [cs]. 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, UK: SAGE Publications.
- 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: 0.1111/j.1467-954X.1984.tb00113.x
- 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 Publishing.
- 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(1), 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.
- 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), 5:1–5:29. 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. (2008). *Software Studies: A Lexicon*. Cambridge, MA: The 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., van der Vlist, F. N., Helmond, A., & 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'*. Artur-Woll-Haus, University of Siegen, Germany. Retrieved from <http://bit.ly/app-support-ecologies>
- 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), 1–11. 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 Publishing.
- Hui, Y. (2016). *On the Existence of Digital Objects*. Minneapolis, MN: 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(2), 77–131. doi: 10.1002/smr.344
- Light, B., Burgess, J., & Duguay, S. (2016). The walk-through method: An approach to the study of apps. *New Media + Society*, 20(3), 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: The MIT Press.
- 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(5), 655–686. doi: 10.1177/0162243915574602
- Matviyenko, S., Ticina Clough, P., & Galloway, A. R. (2015). On Governance, Blackboxing, Measure, Body, Affect and Apps: A conversation with Patricia Ticina 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: The 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: The 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
- 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: Wiley.
- Parks, L., & Starosielski, N. (2015). *Signal Traffic: Critical Studies of Media Infrastructures*. Champaign, IL: 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(1), 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(4), 990–1029. doi: 10.1162/154247603322493212
- Rogers, R. (2013). *Digital Methods*. Cambridge, MA: The MIT Press.
- Suchman, L. (2007). *Human-machine reconfigurations: Plans and situated actions*. Cambridge, UK: Cambridge University Press.
- Vannini, P. (2008). Situatedness. In L. M. Given (Ed.), *The SAGE Encyclopedia of Qualitative Research Methods* (pp. 815–816). Thousand Oaks, CA: SAGE Publications. 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
- Weltevrede, E. (2016). *Repurposing digital methods: The research affordances of platforms and engines* (Ph.D.). University of Amsterdam, Amsterdam. Retrieved from <https://wiki.digitalmethods.net/Dmi/RepurposingDigitalMethods>
- Wilson, F. (2014). *Feature Friday: Checking In*. Retrieved from <https://avc.com/2014/05/feature-friday-checking-in/>
- Woolgar, S. (1990). Configuring the user: the case of usability trials. *The Sociological Review*, 38(S1), 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(1), 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.