



UvA-DARE (Digital Academic Repository)

Digital methods for sensory media research: Toolmaking as a critical technical practice

Chao, J.; van Geenen, D.; Gerlitz, C.; van der Vlist, F.N.

DOI

[10.1177/13548565241226791](https://doi.org/10.1177/13548565241226791)

Publication date

2024

Document Version

Final published version

Published in

Convergence : The International Journal of Research into New Media Technologies

License

CC BY-NC

[Link to publication](#)

Citation for published version (APA):

Chao, J., van Geenen, D., Gerlitz, C., & van der Vlist, F. N. (2024). Digital methods for sensory media research: Toolmaking as a critical technical practice. *Convergence : The International Journal of Research into New Media Technologies*, 30(1), 236–263. <https://doi.org/10.1177/13548565241226791>

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (<https://dare.uva.nl>)

Digital methods for sensory media research: Toolmaking as a critical technical practice

Jason Chao¹ , Daniela van Geenen¹ ,
Carolín Gerlitz¹ and Fernando N van der Vlist² 

¹CRC Media of Cooperation, University of Siegen, Siegen, Germany

²University of Amsterdam, Amsterdam, Netherlands

Convergence: The International
Journal of Research into
New Media Technologies
2024, Vol. 30(1) 236–263
© The Author(s) 2024



Article reuse guidelines:

sagepub.com/journals-permissions
DOI: 10.1177/13548565241226791
journals.sagepub.com/home/con



Abstract

‘Digital methods’ turn to medium-specific and online avenues for social and cultural research. While these approaches foster empirical media studies, it has become increasingly challenging to ‘follow the medium’ and ‘repurpose’ its methods. The prominence of sensory media such as ‘smart’ networked devices (e.g. mobile phones) in mundane practices and their infrastructural dependencies confront media scholars with highly contingent objects of study. Yet, studying such sensor-based devices is crucial, for they enable continuous and unnoticed monitoring of everyday (inter) activity. The article suggests that developing digital methods for sensory media can be understood as specific ‘critical technical practice’ (CTP) by engaging with two toolmaking stories. It draws on and emphasises the fundamental similarity between CTP and digital methods which both aim at conjoining technical engagement and understanding with methodological reflection. The toolmaking stories explicate the making of and the limitations to developing digital methods for increasingly obfuscated mobile sensory media, exploring the possibilities of repurposing their functionality and data. They include building tools for app code analysis focused on apps’ capacity to track sensor data, as well as for ‘sensing’ and analysing network traffic of mobile devices in use. The featured toolmaking then unravels distinctive research affordances, that is, action possibilities for ‘static’ and ‘dynamic’ modes of analysis grappling with the technicity of mobile sensory media and their data. We argue that toolmaking as CTP for sensory media studies implies engaging with these media as entangled infrastructures, examining not just their social, but also their technical ‘multi-situatedness’. This involves investigating the ‘liveliness’ of their data, or how it is generated, processed and made sense of. In conclusion, we discuss implications for ‘doing digital methods’ in sensory media research. Toolmaking itself becomes an inevitable form of media research and critique, inviting and challenging researchers to deploy the media’s situatedness for their investigations.

*Authors contributed equally.

Corresponding author:

Daniela van Geenen, CRC Media of Cooperation, University of Siegen, Herrengarten 3, Siegen 57072, Germany.
Email: daniela.vgeenen@uni-siegen.de

Keywords

digital methods, critical technical practice, mobile sensory media, app studies, toolmaking, toolmaking stories, data critique, media critique

Introduction

The rise of online digital media has been accompanied by researchers from various disciplines who had an interest in repurposing their data for their specific research objectives (e.g. [Savage and Burrows, 2007](#)), using it as insights into preferences, sociality, controversies, but also economic relations and technical operations. Since the mid-2000s, a particular outlook on studying ‘natively digital’ objects using digital data for research has formed under the label of ‘digital methods’ ([Rogers, 2013](#)). This label denotes a media research approach that takes the data, the technical specifications and functionality of digital media – may these be webpages, blogs, wikis, web archives or social media, and users’ links, likes, posts or comments, to name only a few – to study both the social and the media.

While such more web-based approaches have been productive for and shaped empirical media studies in important ways, it is becoming increasingly challenging to ‘follow’ and ‘repurpose’ digital, platformised or mobile media in recent years. The very structure and logic of the web has changed significantly with the proliferation of logins and ‘paywalls’, personalised and localised user experiences on the web, on mobile devices and in apps. Infrastructurally, the web evolved from (more or less) static and ‘open’ websites and pages to increasingly ‘closed’ social media and mobile device architectures, distributed but interrelated social ‘widgets’ (e.g. [Gerlitz and Helmond, 2013](#)) and mobile apps (e.g. [Dieter et al., 2019](#)), connected ‘smart’ devices, including surveillance technologies, autonomous vehicles and voice-controlled ‘intelligent personal assistants’. In fact, it has become challenging to speak of a ‘single medium’ and its specificity, as envisioned in the context of earlier digital methods. On the contrary, the interconnectedness and interdependencies of media technologies, practices and devices in digital media infrastructures lead us to address and approach media in the plural, making an effort to ‘follow the media’ to study their socio-technically entangled quality.¹

Furthermore, leading digital media platforms increasingly restrict access to data and services through their application programming interfaces (APIs) in response to mounting public controversies and concerns such as the Facebook–Cambridge Analytica ‘data scandal’, changing privacy legislation (such as the GDPR), and also for competitive reasons ([Van der Vlist et al., 2021](#)).² Consequently, the way in which platforms and APIs are governed has raised the critical need to look for alternative research approaches suitable for a ‘post-API’ environment ([Perriam et al., 2020](#); [Venturini and Rogers, 2019](#)). Proposals include new forms of ‘digital fieldwork’ ([Venturini and Rogers, 2019](#)), qualitative and mixed methods ([Perriam et al., 2020](#)) and to ‘follow the *natives* (along with the *medium*)’ ([Caliandro, 2021](#)), specific forms of digital ethnography or ‘appnography’ ([Cousineau et al., 2019](#)), ‘walkthrough methods’ ([Light et al., 2016](#)), as well as (empirical) studies of the ‘situatedness’ of mobile apps, app stores and infrastructures ([Dieter et al., 2019](#); [Gerlitz et al., 2019b](#)).

Within this paper, we discuss the consequences of these developments for devising novel digital methods for mobile media and the ‘apps’ that make these media technologies operational (e.g. [Gerlitz et al., 2019a](#)). It is rarely the case that apps (or app versions) have the same appearance or user experience for everyone. Moreover, the infrastructures that apps rely on to function properly, are updated frequently and are characterised by obfuscation ([Dieter et al., 2019](#)). Both data retrieval

and data forms are less structured than HyperText Markup Language (HTML) or data retrieved via APIs. Most notable are the various types of sensory data captured in the background, such as location, movement and ongoing data exchanges. Given all the many different issues and challenges, the question is *whether* and *how* it might still be possible to follow the media and repurpose their methods. Therefore, this paper opens up the question of the future of digital methods for mobile sensory media including mobile devices, wearables, voice assistants, autonomous and connected vehicles, and the ‘internet of things’ more generally. How do we build methods and design tools for such increasingly personalised, localised, interconnected and obfuscating digital media?

We begin approaching this challenge by (re-)turning to the relation between digital methods and ‘critical technical practice’ (CTP), a working ethos proposed by Philip [Agre \(1997a and 1997b\)](#) in the context of artificial intelligence (AI) research and development. Digital methods development has previously been considered to resonate with CTP as a scholarly attitude through iterative method design, in which technical engagement and making go hand in hand with methodological reflection (cf. [Rieder and Röhle, 2017](#)): Digital methods are meant to combine a socio-technical understanding of media, gained through repurposing their data, technical constitution and functionalities, with critical reflection on how certain assumptions and values are inscribed into these media’s methods ([Agre, 1997b](#) in [Rieder and Röhle, 2017](#)). Taking inspiration from CTP, we present two ‘toolmaking stories’, as we term them, which highlight the challenges and possibilities of devising digital methods for smartphones (or other devices with a network connection) and their apps. The first toolmaking story concerns the making of *AppInspect*, developed for inspecting app code, to identify diverse actors in the codebase and scan, for instance, for privacy-invading instructions focussing on sensor data access. In the second toolmaking story, we describe and reflect on the making of *AppTraffic*, built for ‘sensing’, capturing and inspecting the network traffic of mobile devices, which is essential to the proper functioning of all ‘tethered appliances’ ([Zittrain, 2008](#) in [Dieter et al., 2019: 9](#)). As we will demonstrate, CTP in relation to mobile sensory media means engaging with these media’s ‘infrastructural entanglements’, that is, examining the ways in which they are ‘connected to objects and infrastructures without knowing exactly how or where’ ([Mackenzie, 2010: Abstract](#)). Thinking toolmaking as specific CTP will allow us to uncover and outline the obfuscating, but also infrastructured character of mobile sensory media in order to provide the foundation for future mobile and sensory digital methods.

The article is structured in four main sections. We first situate our contribution in the current interdisciplinary debates on sensory media and their investigation and sketch the broader importance of sensory digital methods for empirically invested media studies endeavours. Secondly, we elaborate on our approach to ‘toolmaking’ as CTP for digital methods research. Thirdly, we present the two toolmaking stories on *AppInspect* and *AppTraffic* and reflect on these tools’ development and prototyping processes. These two stories illustrate the differences between what we call *static and dynamic modes of analysis* in the study of mobile sensory media and how these modes of analysis complement each other. Each mode signifies unique ‘research affordances’ ([Weltevrede, 2016](#)), that is, action possibilities digital devices and environments offer – or in our case disclose – for doing empirical research. Finally, we discuss the implications for ‘doing digital methods’ ([Rogers, 2019](#)) in the context of sensory media. We reflect on the scholarly relevance of toolmaking and toolmaking stories, and advocate for a more explicit alignment with CTP to foster empirical media research and media critique through cultivating data and infrastructure literacy.

Sensory media and data critique

The proliferation of sensor technologies in our everyday digital media environment has been approached by many interdisciplinary media scholars, through conceptual and empirical perspectives. Among the first contributions was [Andrejevic and Burdon's \(2015\)](#) claim of an emerging 'sensor society', in which 'interactive', highly responsive, mobile and wearable media technologies allow for ongoing 'sensor-based monitoring' and 'sensor-derived data' (23), which inform decision-making in a growing number of diverse areas of life. These technologies are marked by their ability to continuously – and largely unnoticed – sense human (inter)activity and based on that capture data in the background ([Ibid.](#); [Salter 2022](#)). In the following we focus on contributions that explore the extent to which sensory media and their data are already infrastructurally entangled with mundane practices in social, cultural, economic and political life to outline the challenges and affordances they carry for digital media research and critique.

The extent to which sensory media permeate both public spaces and private living environments is illustrated by sensor-enabled and networked devices constituting the 'internet of things'. [Bunz and Meikle \(2017\)](#), for instance, discuss and problematise how personal and intimate data become economic assets. Sensory media might inform the management of public spaces and transportation, for example, through social navigation (cf. [Gekker and Hind, 2020](#); [Kanderske and Thielmann, 2019](#)), and allow data capture in domestic settings or in personal contexts through 'smart' devices such as intelligent personal assistants ([Pridmore et al., 2019](#)) or self-monitoring apps. The latter rapidly gained societal traction in 2020 due to the COVID pandemic, which aroused extensive public interest in health and mobility monitoring (e.g. [Dieter et al., 2021](#); [Ferretti et al., 2020](#)). Increasing critical attention is paid to the expansion of monitoring sensory media afford, focussing on possible redistribution of societal power relations (e.g. [Crawford, 2021](#)) or their impact on international security infrastructures, including in urban spaces (i.e. 'smart cities') and at borders, through drone and facial recognition technologies and health applications ([Klimburg-Witjes et al., 2021](#)).

Besides such data critique of sensory media, several scholars have explored how their data can be studied empirically. Many approaches focus on experiences and experiments with self-tracking apps as well as exploring sensory data capture associated with the practices using mobile devices. A large part of this work draws on ethnographically informed methodologies to study sensory media and related data practices entangled in specific settings and situations of use. [Pink et al. \(2017\)](#), for example, made a group of self-tracking cyclists record their daily commuting routines and mobility through the devices and apps they normally use in combination with GoPro camera footage. The captured data was contextualised through follow-up interviews to discuss and understand the cyclists' data practices. The researchers coined the notion of 'mundane data' to advocate the need for an understanding of data produced through everyday activity and routines ([Pink et al., 2017](#)), in which humans are actively generating, navigating and making sense of data. They state '[t]he mundane is a site where data are lively' ([Pink et al., 2017](#): 4), in which their understanding of 'lively data' ([Lupton, 2016](#)) concentrates on addressing the situated, embodied and affective dimensions of the data practices involved in self-tracking activities. In a similar fashion, [Nafus \(2018\)](#) refers to 'the social life of methods' (amongst others, [Ruppert et al. \(2013\)](#)) to stress that for the sensemaking of sensory media and their data the devised and employed methodological approaches should be tied to an understanding of the 'ongoing social lives that they are designed to comprehend' (235–236). Furthermore, Nafus aptly notes that 'taking the social life of methods seriously also means intervening in the tools that sustain cultures of big data' ([Nafus, 2018](#): 236). Similarly, Rettberg, asks

to focus on the situated nature of sensory data in everyday use practice and social relations, taking self-tracking technologies, especially the Strava app, as primary example (Rettberg, 2020).

While these authors focus on the social embeddedness of sensory media and data, others highlight the ever-shifting computational infrastructures and their underlying political economies that facilitate and frame the continuous ‘making’ of digital data, advocating for ‘data infrastructure literacy’ (Gray et al., 2018). In their exploration of mobile apps, Dieter et al. (2019) propose a more comprehensive understanding of ‘lively data’ than just a social reading of this term and suggest that mobile apps and their data are ‘multi-situated’ in terms of location, practice, infrastructure and technicity, emphasising the need for a thorough and ongoing technical examination of sensory media in action. Here, situatedness incorporates the unstable technical conditions and infrastructural settings that bring about and (further) process digital data of sensory media, which one needs to come to terms with in order to develop data infrastructure literacy in this context.

Sensor-enabled devices owe their specific functionality to the processing of data generated through such sensors. For instance, these devices rely on hardware motion sensors such as the built-in accelerometer for orientation and movement, and the gyroscope used for the autorotation of the view on the device screen.³ Yet, the resulting measurements are neither simply accessed and made sense of by either users or researchers without technical knowledge and equipment, nor are they usually presented to them in the front-ends. A series of contributions offers empirical entry points to study such technical-infrastructure situatedness of mobile sensory media, largely by combining ‘technical fieldwork’ with the ‘hacking’ and (reverse) engineering of the workings and entanglements of such technologies (e.g. Blanke and Pybus, 2020; Dieter et al., 2019; Gerlitz et al., 2019a; Pybus and Coté, 2021). This research has shown that, in contrast to web-based data sources, sensory data processing is often obfuscated and dispersed across infrastructure, and rarely accessible or observable, let alone directly comprehensible. Such data can be encountered in many different types, and in their aggregate forms the data is filtered, further processed and calculated after the event with respect to the relevant ‘features’ to be ‘detected’, ‘recognised’ and ‘extracted’ (cf. Tickoo and Iyer, 2017). That is, the utility value of mobile sensory data could be classified as ‘operational’ rather than ‘representational’ (cf. Rettberg, 2020). Unlike data sets provided by social media APIs, mobile sensory data is rarely used – or useful – as single data points and its analysis cannot easily be performed without knowledge of its context of production, as the empirical examples in this article will underscore. Also, mobile sensory data is uniquely tied to extensive ‘cloud’ computing infrastructures for purposes of data storage, aggregation, processing and analytics in particular (cf. Dieter et al., 2019; Gerlitz et al., 2019a). Thus, the critical examination of sensory media asks for a practical engagement with their technicity, that is, their technical specifications and workings, including an exploration of required network connections. In the next section, we argue and demonstrate that ‘toolmaking’ offers a crucial entry point for such a critical technical engagement.

Toolmaking stories

While research methodologies and their challenges are commonly given critical attention, computational tools are not always approached with and might not even invite the same reflexive scrutiny (Rieder and Röhle, 2017; Ruppert et al., 2013; Van Geenen, 2020). As Rieder and Röhle (2017) have pointed out, our digital tools thrive on and ‘mobilise’ (111), but are also ‘black-boxing’ (112), built-in methods imported from a variety of disciplines and sectors, of which the influence on the research results could effortlessly be disregarded. In the humanities and social sciences one can even witness an increasing divide between tool developer and tool user (cf. Van Geenen, 2020; see also Rieder et al., 2022). The capacity to investigate sensory media is linked to research strategies

and tools able to address, dissolve or circumvent the infrastructural obfuscation of sensory data capture and processing discussed above. ‘Toolmaking’ then becomes a vital part of knowledge production, an active and ongoing process through which (purposefully) hidden or invisible computational operations can be rendered accessible and assessable. In that sense, our emphasis on toolmaking is meant to integrate ‘the craft work of design’ and the ‘reflexive work of critique’, as suggested by [Agre \(1997b: 155\)](#). Especially when approaching emerging media, such as mobile and sensory media, toolmaking relies on but also expands researchers’ socio-technical understanding of the media. Accounting for the challenges of making tools thus contributes to both a methodological sensitivity and, as we seek to show, media critique.

In its original conceptualisation, Agre devised CTP to encourage practitioners in the area of AI research and development to investigate and render visible the theoretical underpinnings, methodological concepts, social norms and cultural values built into and practised through computational technologies (cf. [Agre, 1997a](#)). Agre’s conceptualisation of CTP has been described more broadly as a form of critique in technical practice as part of and invested in ‘situated action’, in which paying ‘attention to the rhetorics and technologies through which a field constructs its research objects becomes an integral part of its research practice’ ([Suchman, 2007: 14](#)). In short, we understand ‘toolmaking’ as a vital part of digital methods development for sensory media studies and as a specific form of CTP combining the making and repurposing of technology with the consideration of its built-in values, its societal implications and infrastructural entanglement. The toolmaking this article encourages aims for tools that promote critical technical engagement, understanding of and intervention in mobile sensory media.

Rather than addressing solely practitioners from engineering fields or with otherwise extensive and advanced technical knowledge, we suggest that CTP has epistemological significance for a wider media studies audience and propose ‘toolmaking stories’ as a way to explicate the role of tools – and thus technical work – in media research and critique, in knowledge production and methodology design. We strive to demonstrate that this textual genre offers opportunities to account for the ‘cascades of inscriptions’ ([Ruppert et al., 2013: 31](#)) that mobile sensory media and tools to access them feature and that are reflected in these media’s data. Such inscriptions result from situated activities of and interactions between users – including us, as scholarly practitioners – mobile devices and their built-in hardware sensors, apps, operating systems, as well as distributed computing infrastructures and services, which all contribute to these sensory media’s performative materiality.

In introducing the ‘hybrid genre’ (cf. [Agre, 1997a: XIII](#)) of ‘toolmaking stories’ into media studies research, we hope to encourage fellow researchers and practitioners to write down and share their toolmaking challenges and reflections, not only after completion of projects and publication of results, but accompanying the research process.⁴ We conceive of ‘toolmaking stories’ as opportunity for (scholarly) practitioners to provide insights in the ‘realities of toolmaking, where decisions have to be made at every step of the way [...] that will affect those who use these tools and, by extension, the knowledge that is produced with their help’ ([Rieder et al., 2022](#)). While there can be examples found of such an academic writing practice (e.g. [Jacomy et al., 2014](#); [Rieder, 2013](#); [Wieringa et al., 2019](#)), it is not yet a common part of scholarly conduct or a fully-fledged textual style in the humanities and social sciences. Yet, scholarly investigations as cooperative practice, especially in digital media research, where different disciplines collaborate, benefit from toolmaking stories. As we will show, the following toolmaking stories allow for two valuable perspectives on studying sensory media, taking the idea of ‘critique’ in CTPs further: firstly, to build tools in order to enquire into mobile sensory media, one needs to understand – and can thus assess – their infrastructural

entanglements; and secondly, tool design is being accounted for as a knowledge practice in empirical digital media research.

In the two toolmaking stories that we present, we build on prior experiments with studying Android and iOS-powered mobile devices, mobile apps and the infrastructure they require to operate (e.g. Dieter et al., 2019; Gerlitz et al., 2019a). In these cases, researchers were confronted with the question of how access to built-in hardware and software sensors is mediated and governed by the operating systems through APIs and application permissions that users accept or decline. In addition, the toolmaking stories draw from previous research concerning data flows and global data infrastructures of mobile sensory media – and the difficulty of their capture (cf. Weltevrede and Jansen, 2019; Wilken et al., 2019) – as well as the distributed nature of app source code, that builds on many external software libraries and frameworks (cf. Binns et al., 2018; Blanke and Pybus, 2020). In a next step, we show how we use these initial studies as a starting point for our own toolmaking, developing and prototyping sensory digital methods and tools in iterative, but not always straightforward ways.

The toolmaking of *AppInspect* and *AppTraffic*

The following two toolmaking stories emerged from a larger research strand interested in mobile and sensory media (Dieter et al., 2019; Gerlitz et al., 2019a) enquiring into the ways in which such media and their apps are entangled in data collection and exchanges. Key research questions that arose from this research strand revolved around what data mobile sensory media do get access to and where they share that data. As a response to these research questions, we set out to develop tools that would repurpose the functionality of mobile media and their data for research. The first toolmaking story we present in this article approaches the question of how we could use methods of software and app analysis to identify whether apps gain access to sensor data on mobile devices, leading to the development of the tool *AppInspect*, which offers access to the enclosed infrastructures of app software. It uses the computational infrastructures of app software packages such as the Android Package Kit (APKs) and offers a static mode of analysis (see Figure 1). The second toolmaking story explores possibilities to ‘sense’ the data mobile devices and their apps exchange in operation, leading to the development of *AppTraffic*, a tool for real-time capture of data flows on mobile devices. It accounts for the actual data flows in use and thus offers a dynamic mode of analysis (see Figure 1). In both stories, we will provide an account of and reflect on the technical affordances, values at stake and challenges of making such research tools and demonstrate that technical making can foster modalities of media critique. Our approach towards toolmaking is usually a collaborative and interdisciplinary one, as tools are designed as part of ongoing research efforts by researchers who combine a media studies expertise with skills in coding, knowledge of data infrastructures and tool development. Both presented tools have been created by Jason Chao as are part of his PhD in media studies, in close and iterative cooperation with the group of co-authors and other scholars affiliated with the App Studies Initiative.⁵

Studying the data apps can access – the making of AppInspect

The question of what data apps get access to, has been central to many privacy-aware studies (Blanke and Pybus, 2020; Pybus and Coté, 2021). Some kinds of data are considered to be more sensitive than others, such as user or device IDs, but also sensor data, that is, data about location, movement or bodily activities, which we are interested in. Researchers have, for instance, trained machine learning models on a wide range of smartphone sensors (including the above-mentioned

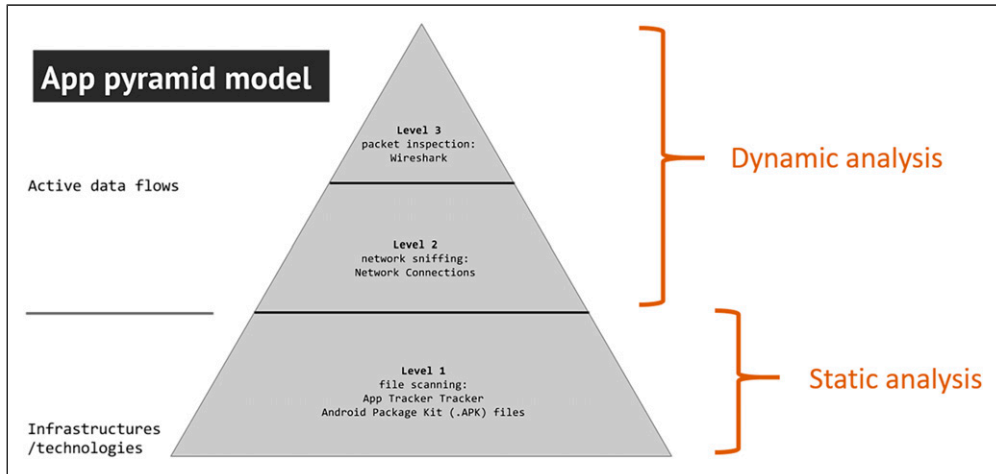


Figure 1. Adapted version of the ‘app pyramid model’ (Helmond et al., 2018) with ‘static’ and ‘dynamic’ modes of analysis.

accelerometer and gyroscope) to recognise human activity and predict the users’ actions (such as walking, running, and cycling.) (e.g. Chen et al., 2021). Our research objective was to identify ways to study whether and to what degree apps gain access to different sensor data when installed on a mobile device. In the context of the web, much research has drawn on code analysis to identify which data is collected by which tracker, widget, or snippet. Our aim was to develop an approach that would allow for app code inspection. App code, however, is much more difficult to access as it is infrastructurally more enclosed in modular app packages (e.g. APKs) and there are (as of yet) less predefined modes of analysis. In order to develop a method and a supporting tool, we first needed to understand how sensor access is organised in technical-infrastructural ways.

From the perspective of user experience, app permissions appeared to be a reasonable starting point: Downloaded apps ask users during the installation process to give permission to access specific sensors for functionality – and in fact, sometimes not just for the sake of functionality.⁶ If permissions are granted, they provide the app with ongoing access to ‘listen’⁷ to the specific device sensors on a smartphone, for example, provoked by app use, but this relationship remains mediated and controlled by the operating system. Can one identify the access to sensor data by looking at the permissions users need to grant an app when installing it? It is a first starting point, but previous research by Pybus and Coté (2021) has shown that such access can also be invoked without asking for explicit permission. The authors demonstrate that permissions do not provide all information about the access to or generation of the diverse sensor data apps can invoke. They show that more insights can be gained from the so-called ‘AppManifest’, a document required by app stores and accessible within the software of an app (i.e. the app package). For instance, Android Application Package (APK) files consist of the AndroidManifest.xml file, the app’s (compiled) code, software ‘libraries’, ‘classes’, ‘resources’, and ‘assets’. As apps increasingly include third-party functionalities, such as social media platform integration or pre-programmed libraries, the APK contains both first- and third-party code.⁸

Here, we encountered a key difference between online and app research: while the source code of websites can be inspected in browsers’ developer tools, app packages are not as easily accessible

and introduce an infrastructural obfuscation (Dieter et al., 2019). They need to be obtained and ‘decompiled’ to read their code, which in the case of iOS is subject to restrictive platform governance by Apple, resulting in our focus on Android APKs in the following.

To make sense of this type of code, we firstly conducted a development experiment and built a simple Android application that reads and shows the data from the compass, accelerometer and gyroscope in real-time. By building an app that draws on sensor data, we sought to identify where the access to such data on the device is invoked on a technical level. Interestingly, we discovered that neither the permissions nor Manifest file of this experimental application does contain any sign of these types of sensor data being retrieved, which was alerting us to expand our code analysis further. A closer look into the developer documentation for Android apps shows that only specific types of sensor data require a declaration of permissions or runtime features in the Manifest file, including: geolocation, microphone, camera, Bluetooth (i.e. the IDs of nearby Bluetooth devices), Wi-Fi (such as names of nearby Wi-Fi hotspots) and NFC (e.g. names of radio frequency identification tags). Google even recommends developers a practice that circumvents listing access to data in permissions and the Manifest by invoking the ‘pre-installed camera app’ to capture images and videos (Android Developers, 2023). In this way, an app’s retrieval of sensor data could be delegated to the device manufacturer’s app, structured as API calls to read media data from an external app.⁹ Therefore, neither app permissions nor the AppManifest are sufficient when exploring sensor data access, which is an insight that we could only gain by creating an app development situation, looking at the app’s infrastructural embedding as a developer. This finding is not just interesting regarding our toolmaking efforts, but also from a privacy-critical perspective, as it shows that there is a technical possibility to access device sensors without making that explicit to users through permissions or in the more technical layer of the manifest.

Having explored the different dimensions through which sensor access is infrastructurally enabled, we decided to build a tool for app code analysis that would enable non-technical media scholars to gain access to app code of larger collections of apps by creating APK libraries and searching the code for specific functions. By doing so, the tool aims at facilitating analytical access to the enclosed code environment of apps. For the detection of sensor data access, we decided to focus on above-mentioned API calls to sensor data for our follow-up toolmaking by means of borrowing static code analysis techniques from malware detection in computer sciences (Wu et al., 2016; Zhang et al., 2019). A first prototype version emerged and was named *AppInspect* (Chao, 2023a, 2023b, 2023c) – an app code analysis tool envisioned to systematically download, store, query and analyse app packages (see Figure 2). The analytical features contain data types defined in the app packages, such as permissions, the presence of trackers or the use of API calls as detailed by the device manufacturer. As app packages consist of developed and built-in third-party code, *AppInspect* allows to identify where in the code certain features have been implemented and enables to connect them to their originating developer. Following a modular development approach, more analytical features can be added based on future researchers’ needs. In 2020, the tool was tested by some of us in the context of a research project on COVID-19 response apps, asking whether these apps gain access to sensor data on the devices installed.

For this purpose, a set of official international COVID-19-response Android apps downloaded in Summer 2020 (Figure 3) has been screened for the Android API call ‘SensorManager.>registerListener’ in the code files. The purpose of an app’s invocation of ‘SensorManager.>registerListener’ is to request the Android system to constantly update the app on any change in the value of measurements by a device sensor.¹⁰ This instruction may be seen as a prerequisite for getting sensor data. The experiment showed, for example, that the Australian government’s check-in app contains nine places getting sensor data, four of which we assume are written by the app’s

publisher. As *AppInspect* also allows to screen the integrated third-party code of apps, such as for the presence of social media integration and its code, the tool detected the other five places in Facebook’s and Google’s software development kits (SDKs) providing these platforms with sensor data access.

A limit to the approach of simple API call scanning is that we cannot tell precisely what types of sensor data are retrieved – disclosed is merely the infrastructural possibility of data access. The request for data from a specific sensor is passed to the API call as a parameter. Once an API connection is initiated, the app can continuously receive updates on changes in the sensors’ readings. We tried to use reverse engineering techniques to explore an automatic and scalable way to trace the line of code where the parameter specifying the sensor would be set. This, however, appeared to be overly technically demanding and time-consuming as each application and SDK is coded very differently. Therefore, at this stage, we focused on the technical-infrastructural condition

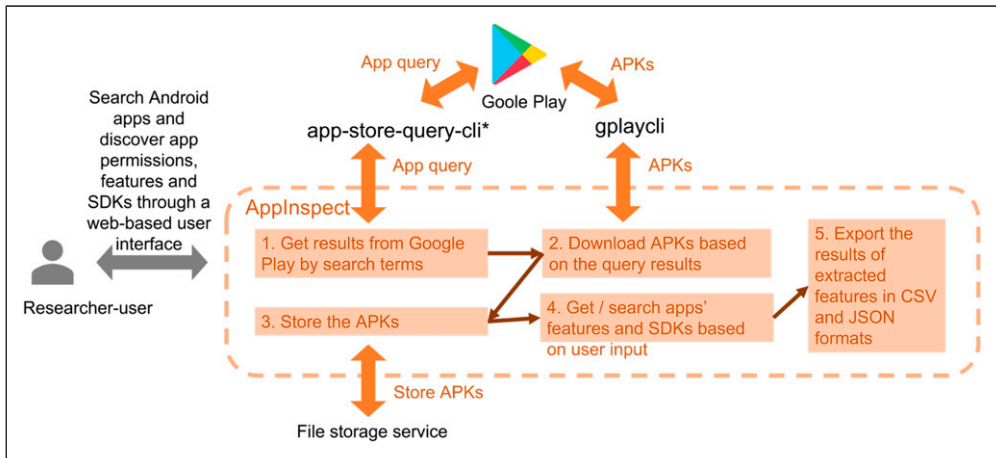


Figure 2. Concept diagram of *AppInspect*. Note: Tools marked with an asterisk are developed by the authors.

```

1 au.gov.act.health.checkin/smali/com/facebook/react/common/ShakeDetector.smali: invoke-virtual (p1, p0, v0, v1),
  Landroid/hardware/SensorManager; ->registerListener (Landroid/hardware/SensorEventListener; Landroid/hardware/Sensor;)Z
2 au.gov.act.health.checkin/smali/com/facebook/appevents/codeless/CodelessManager.smali: invoke-virtual (p0, v1, v0, v3),
  Landroid/hardware/SensorManager; ->registerListener (Landroid/hardware/SensorEventListener; Landroid/hardware/Sensor;)Z
3 au.gov.act.health.checkin/smali_classes2/expo/modules/location/LocationModule.smali: invoke-virtual (v0, p0, v1, v2),
  Landroid/hardware/SensorManager; ->registerListener (Landroid/hardware/SensorEventListener; Landroid/hardware/Sensor;)Z
4 au.gov.act.health.checkin/smali_classes2/expo/modules/location/LocationModule.smali: invoke-virtual (v0, p0, v1, v2),
  Landroid/hardware/SensorManager; ->registerListener (Landroid/hardware/SensorEventListener; Landroid/hardware/Sensor;)Z
5 au.gov.act.health.checkin/smali_classes2/expo/modules/sensors/services/BaseSensorService.smali: invoke-virtual (v1, p0, v0, v2),
  Landroid/hardware/SensorManager; ->registerListener (Landroid/hardware/SensorEventListener; Landroid/hardware/Sensor;)Z
6 au.gov.act.health.checkin/smali_classes2/host/exp/exponent/p/v/g/c.smali: invoke-virtual (v1, p0, v0, v2),
  Landroid/hardware/SensorManager; ->registerListener (Landroid/hardware/SensorEventListener; Landroid/hardware/Sensor;)Z
7 au.gov.act.health.checkin/smali_classes2/host/exp/exponent/t/1.smali: invoke-virtual (p1, p0, v0, v1),
  Landroid/hardware/SensorManager; ->registerListener (Landroid/hardware/SensorEventListener; Landroid/hardware/Sensor;)Z
8 au.gov.act.health.checkin/smali_classes2/com/google/android/gms/internal/ads/gr.smali: invoke-virtual (v1, p0, v0, v3, v2),
  Landroid/hardware/SensorManager; ->registerListener (Landroid/hardware/SensorEventListener; Landroid/hardware/Sensor; Landroid/os/Handler;)Z
9 au.gov.act.health.checkin/smali_assets/audience_network/com/facebook/ads/redexgen/X/ls.smali: invoke-virtual (v2, p0, v1, v0),
  Landroid/hardware/SensorManager; ->registerListener (Landroid/hardware/SensorEventListener; Landroid/hardware/Sensor;)Z
10 au.gov.act.health.checkin/smali_assets/audience_network/com/facebook/ads/redexgen/X/r.smali: invoke-virtual (v2, v1, v0, v4),
  Landroid/hardware/SensorManager; ->registerListener (Landroid/hardware/SensorEventListener; Landroid/hardware/Sensor;)Z
11 au.gov.act.health.checkin/smali_assets/audience_network/com/facebook/ads/redexgen/X/r.smali: invoke-virtual (v2, v1, v0, v4),
  Landroid/hardware/SensorManager; ->registerListener (Landroid/hardware/SensorEventListener; Landroid/hardware/Sensor;)Z
12 au.gov.act.health.checkin/smali_assets/audience_network/com/facebook/ads/redexgen/X/go.smali: invoke-virtual (v3, v2, v4, v1, v0),
  Landroid/hardware/SensorManager; ->registerListener (Landroid/hardware/SensorEventListener; Landroid/hardware/Sensor; Landroid/os/Handler;)Z
13 be.sciensano.coronalert/smali_classes2/com/google/zxing/client/android/AmbientLightManager.smali: invoke-virtual (v0, p0, v1, v2),
  Landroid/hardware/SensorManager; ->registerListener (Landroid/hardware/SensorEventListener; Landroid/hardware/Sensor;)Z
14 br.com.novatech.monitoracoronavirus/smali/com/facebook/react/common/ShakeDetector.smali: invoke-virtual (p1, p0, v0, v1),
  Landroid/hardware/SensorManager; ->registerListener (Landroid/hardware/SensorEventListener; Landroid/hardware/Sensor;)Z
  
```

Figure 3. App code scan for sensor-related API calls (i.e. SensorEventListener matches) in decompiled APK files.

of possibility of accessing and reading sensor data in general, before finding a way to detect which specific types of sensor data might be retrieved by particular actors.

In a second development step, a historical dimension was built into *AppInspect*, enabling users to search, download and store historical versions of apps and their APKs in order to study how app code has changed over time. In historical terms, *AppInspect*'s interface allows for the comparison of the number and type of permissions over time. It also makes it possible to explore if the permissions of apps were classified as 'dangerous' according to the Android developer's documentation, repurposing the classificatory scheme of the development environment. In another iteration of experiments, we explored these historical features, this time in the context of a research project interested in the use of sensor(y) data featured in commercial activity tracking,¹¹ focussing on apps like Strava, Runtastic and Fitbit. In order to do so, we first had to identify and cross-check available historical APKs¹² per app and upload them into *AppInspect*. To be more specific, we experimented with historical APK analysis and comparative app analysis focussing on permissions and therein sensor data access.

We found, for instance, that the number of permissions which are classified as 'dangerous' by the Android developer's documentation in the Strava app increased between 2016 and 2022 (from 7 to 11) (cf. [Figure 4\(a\) and \(b\)](#)).¹³ When comparing different apps or historical versions of apps, *AppInspect* first shows the overall counter of permissions classified as 'dangerous' ([Figure 4\(a\)](#)). For further analysis, those can be unpacked by engaging with the details of permissions per version ([Figure 4\(b\)](#)) or by downloading all classified permissions of all apps, enabling users to constantly change the scope of analysis from individual apps to groups of apps, and back.

'Dangerous' permissions' are defined as 'higher-risk permission that would give a requesting application access to private user data or control over the device that can negatively impact the user' ([Android Developers, 2022a](#)), and therefore need to be declared by app developers. When exploring the permissions classified as dangerous in the context of our experiment on activity trackers, most concerned location access and access to body sensors, the performance of activity recognition or bluetooth access.

Yet, in order to evaluate if a permission (e.g. a certain category of sensor data access and retrieval) might be disproportional in terms of an app's functionality, more contextual knowledge and comparison between apps within the same app category (e.g. 'activity tracking') is needed: Next to the already mentioned frequently-occurring 'dangerous permissions', the Fitbit app, for example, also asks for a range of additional ones that enable the Fitbit device to access and write into the calendar, read the call log, make calls, or write sms. These are technical specifications with a privacy-invading character, which might contribute to an ostensibly 'seamless' convenience of use, but that are not strictly necessary for a proper functioning of the device as activity tracker. They also seem to be in conflict with the developers pages' call to 'minimize your permission requests' ([Android Developers, 2023](#)).

Other code features are presented in a similar way in *AppInspect*, for instance, the presence of `EventListener` in the code of one of Runtastic's most recent APKs (see [Figure 5](#)). Here, additional information is provided about 'inferred developers', that is, potential third-party code that also draws on sensor access. Strikingly, scanning Runtastic's app code in this way, we noticed that the app has access to a 'Step.Sensor', which seems to be independent of the declaration of body sensor access or activity recognition functionality. Finding straightforward information on how such step counts are measured, calculated and used seems to be complicated. This marks one of the key and yet unresolved challenges of app code analysis: linking the pre-structured data categories to developer documentation about their functionality and an understanding of what certain permissions, functionalities and processes actually enact, what data they capture or produce.

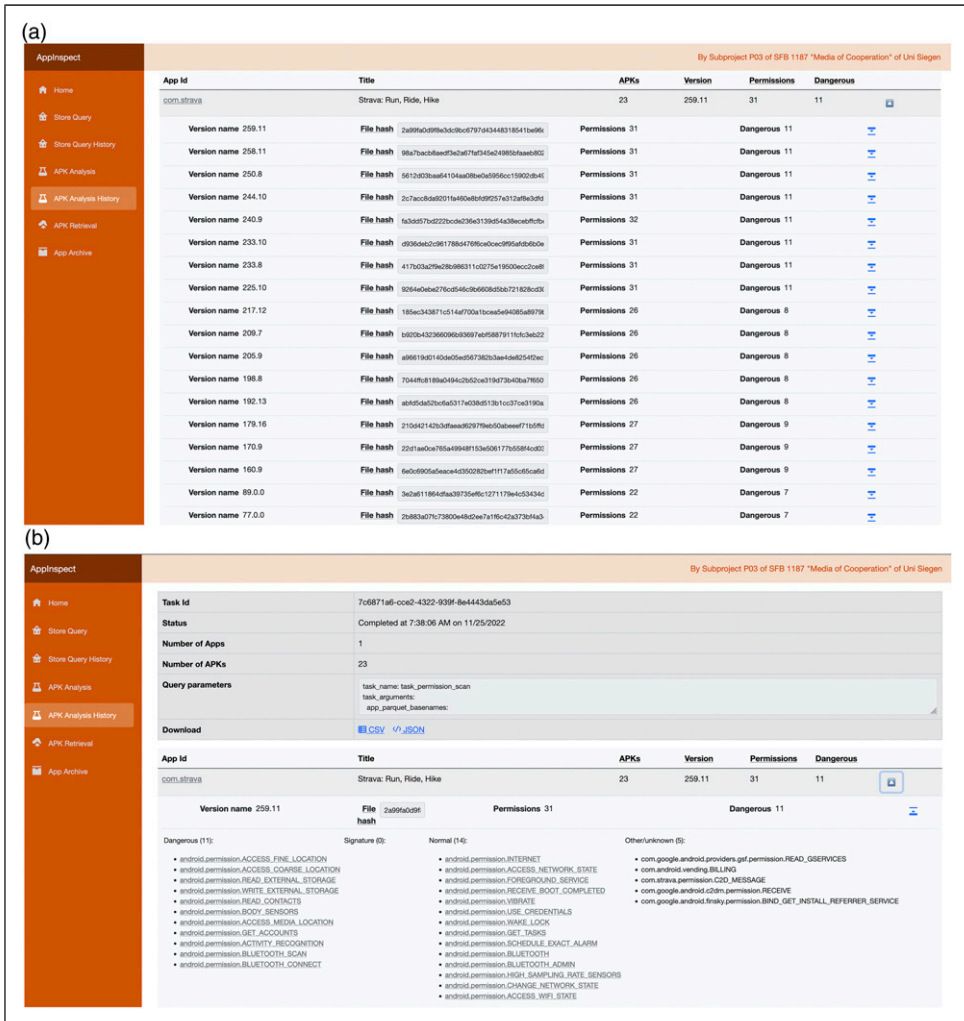


Figure 4. (a) and (b). APK Analysis (History) with *AppInspect* identifying permissions with the option to download the tabular overview per APK as .CSV/.JSON-file.

In summary, the example of prototyping *AppInspect* underscores that building tools cannot be disentangled from understanding how data is generated, ‘infrastructured’ and used in practice. Due to the infrastructural obfuscation of apps and their ecosystems, we could not simply follow these media, as digital methods suggest to do regarding online media. Rather, we needed to create new situations to unpack these media differently, bringing together technical practice and critique of infrastructurally entangled and (partially) opaque sensor data collection. Static analysis approaches can offer one way to study how apps operate as a vital part of sensory media and our initial experiments already indicate that a multiplicity of actors (i.e. platforms, operating systems, third parties) are involved in this process. Toolmaking and its stories are deeply entangled with researching, understanding, and, potentially, critiquing sensory media. *AppInspect* and its toolmaking

App Id	Title	APKs	Version	Developers	Classes
com.runtastic.android	adidas Running: Sports Tracker	2	12.21	2	11
Version 12.21 name	File hash	e5ee70347a68204a522e2ad05fa17afc017426c5		Inferred 2 developers	Matching 11 classes
Inferred developers:	Classes/packages:	Found in code lines:			
<ul style="list-style-type: none"> runtastic google 	<ul style="list-style-type: none"> com.google.zxing.client.android.AmbientLightManager com.runtastic.android.sensor.steps.StepSensor com.runtastic.android.sensor.steps.StepSensor\$1 td.d\$b od.j k9.m k9.s k9.r k9.n com.google.android.exoplayer2.video.spherical.OrientationListener com.google.android.exoplayer2.video.spherical.SphericalGLSurfaceView 	<pre> com.google.zxing.client.android.AmbientLightManager: implements Landroid/hardware /SensorEventListener; invoke-virtual (v0, p0), Landroid/hardware /SensorManager; ->unregisterListener(Landroid /hardware/SensorEventListener;)V com.runtastic.android.sensor.steps.StepSensor: .field private final listener:Landroid/hardware /SensorEventListener; </pre>			

Figure 5. App code scan with *AppInspect* building `SensorEventListener` matches in decompiled APK files.

story offers a first step to understand mobile devices and their apps as part of wider data infrastructures that make them operational as sensory media.

Examining app use ‘in the wild’ – the making of AppTraffic

While the static modes of analysis *AppInspect* features allow us to explore the infrastructural possibilities of apps getting access to sensor(y) data, it leaves the question about the actual data flows enabled by mobile media untackled. Static analysis may provide first clues about external connections apps draw on. Yet, the more complete approach is to capture them ‘live’ making use of ‘dynamic analysis’ once these connections are enabled in mundane settings and specific situations, recording as many connections as triggered by specific conditions, events, locations or other cues. As we have seen in our previous engagement with app code, like web code it is informed by the integration of third-party code and data exchange with third-party actors. We were interested if we could trace these relations through actual data exchanges.

An established starting point to such a question is tracing and analysing so-called network connections, that is, the data in – and outflows between external sources and mobile devices. Sending and receiving data is fundamental to the basic functionalities of mobile devices, including for updates, logins or content retrieval, but it also entails ‘in-app’ advertisements and implies sharing data – including specific sensor data – with third parties. There is a multiplicity of privacy-aware applications that seek to make visible to which external servers a mobile device or even an app connects, such as the app ‘Network Connections’ ([Professional American Service System Corp LLC., 2020](#)). As discussed elsewhere, ‘network connections are a key entry point for understanding how apps are always and necessarily bound up with – or “tethered” to – other objects and infrastructures’ ([Dieter et al., 2019](#): 9 in reference to [Zittrain, 2008](#); see also [Gerlitz et al., 2019a](#)). That is to say, studying network connections figures as a central vantage point for studying the data infrastructures of sensory media, although not without limitations as the following will show.

In the past, network connections were studied by creating laboratory settings, within which all incoming and outgoing data of a device are rerouted through a static, nearby Wi-Fi network capturing them with network sniffers or decryption tools such as *Wireshark*¹⁴ or *TCPDump*¹⁵ (e.g. [Weltevrede and Jansen, 2019](#)). For such an approach to be successful, researchers need to have

control over a network infrastructure and possess the technical knowledge required to change network rules and capture the traffic data with suitable tools.¹⁶ In that sense, such a methodology is both ‘live’, that is capturing data in the moment of its occurrence, and ‘lively’ (cf. Marres and Weltevrede, 2013) as it dynamically adapts to the socio-technical conditions of the media. The existing tools (e.g. *Wireshark* and *TCPDump*) do not accommodate researchers from non-technical backgrounds who are interested in the data flows of apps due to the level of technical knowledge required to configure and operate the tools.

As we had an interest in situated sensing of data flows – especially for the case of mobile devices – we set out to develop a tool that would first make network connection research possible to non-technical media researchers. Second, the tool should allow this investigation to be performed anywhere, thus, ‘in the wild’ and not in a laboratory setting, as is the intended use context of mobile devices – to be on the move. Additionally, the tool needed to be scalable to be used by multiple researchers. The result is a technically well-documented and -maintained app called *AppTraffic* (Chao, 2023d) that draws from the technical capacities of *Wireshark* and makes them usable in the wild, by any researcher and by any number of researchers. Users can download *AppTraffic* on their mobile devices to enable capture sessions through a server-based infrastructure.

In doing so, we first needed to understand how network data gets produced and is infrastructured in order to devise a tool to ‘sense’ that data. The notion of ‘the sensory’ in this part is thus not just concerned with the sensory capacities of the media under scrutiny, but also with that of research methods. This is also why we at several occasions in this article speak of sensory data instead of strictly sensor data. The initial challenge to address in devising *AppTraffic* was the observation that network connections are usually decrypted for privacy reasons. In previous studies, entities like trackers, which are responsible for many mobile data flows, were found to send out users’ private data without encryption using Hypertext Transfer Protocol (HTTP) (cf. Weltevrede and Jansen, 2019). At the time of writing, it is hard to find code entities that send out data without encryption. Attempts to decrypt Hypertext Transfer Protocol Secure (HTTPS) traffic inevitably encounter strong ‘platform resistance’ (Dieter et al., 2019: 10). The operating systems and applications have measures in place to safeguard against the breach of encryption, which is relevant to end-users but presents challenges for researchers. The common approach to decrypting such traffic of mobile devices is to build a research app that instructs the device to trust the decryption process on the device level. This is realised by setting up a Certificate Authority (Android Developers, 2022b; Apple Support, 2022), in order to instruct the device to allow for encryption. It is important to note, though, that not all systems might support this approach.¹⁷ When setting up *AppTraffic*, users are first guided through installing the Certificate Authority on their mobile phone, and thus, are confronted with the fact that they interfere in the hardware resistances to tracking.

To capture network connections, one needs to route them through a capture infrastructure and try to decrypt them. We accomplished this by setting up a virtual private network (VPN) with an associated server where the traffic is rerouted to, recorded and decrypted (Figures 6 and 7). As such a capture process is data-intensive and potentially invasive in user privacy, a development requirement was to give tool users maximum control over the enabling and disabling of such captures. Captures can be enabled on the mobile device in the *AppTraffic* app and can be monitored in a web-interface detailing all captured data.

Considering the diverse (possible) backgrounds of tool users, our requirement was that *AppTraffic* must be as easy to use and as intuitive as possible. Therefore, the tool adopts the ‘wizard design pattern’, which provides a step-by-step process guiding users to accomplish a task, (Babich, 2017) – in our case, creating a capture session of network traffic (Figure 6): The ‘Record’ section in the web-based interface of *AppTraffic* guides the researchers through the steps to pick a capture

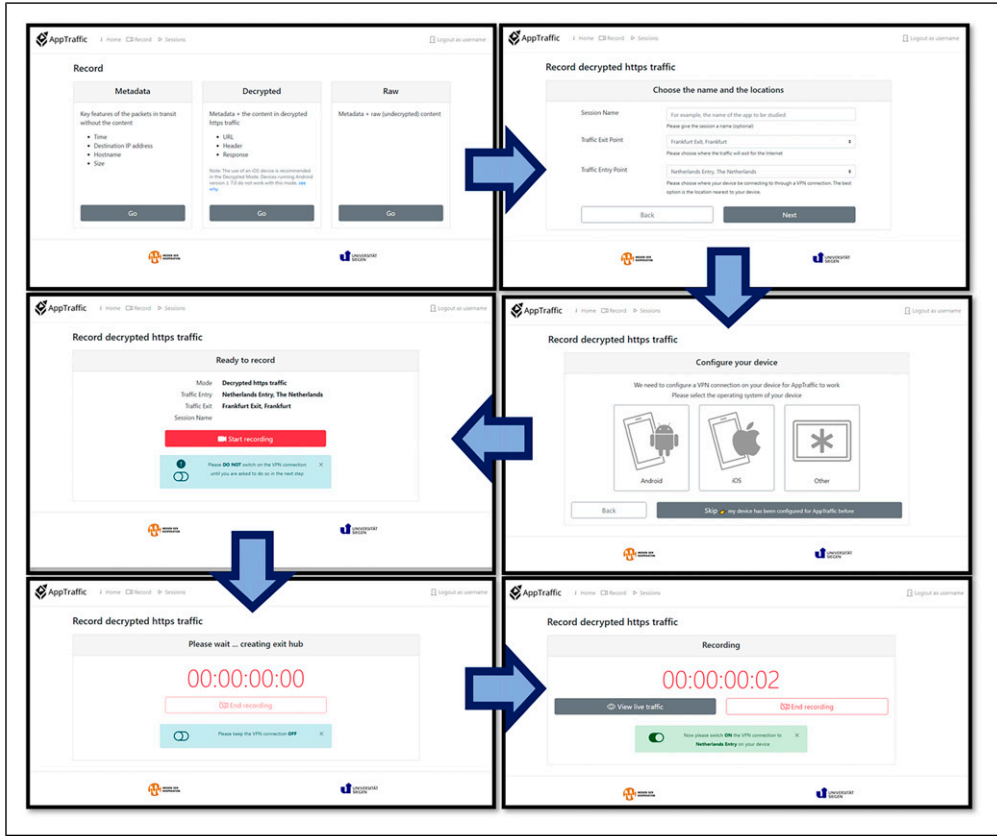


Figure 6. Step-by-step 'wizard' for starting a network traffic capture session with AppTraffic.

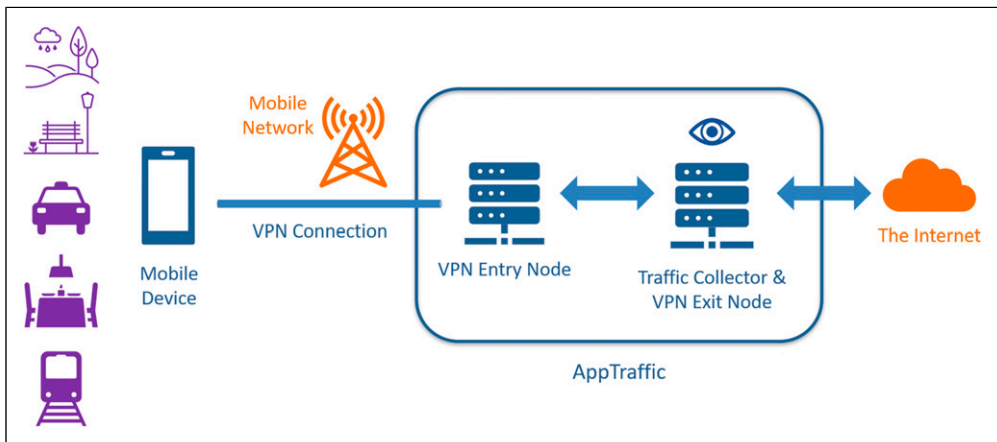


Figure 7. Concept diagram of AppTraffic. A mobile (research) device establishes a VPN connection with AppTraffic over a mobile network.

mode, choose the exit and entry points, configure the mobile device, start recording and turn on the VPN connection.

The use of the VPN (Figure 7) allows capturing network connections in the wild and sensitises researchers for the relevance of situatedness in this type of research: On the one hand, the setup technically works without a comparable use situation and therefore might make data capture more difficult to compare. On the other hand, it does allow researchers to attend to use situations as they unfold on the move and in specific locations, as opposed to controlled situations in the lab (cf. Savage and Burrows, 2007). The setup also supports establishing multiple VPN entry and exit servers so that researchers can simulate using their mobile devices from different locations, and through different activities, and compare network connections across countries. Here, our toolmaking taps into a longstanding debate on laboratory settings and laboratory knowledge in science and technology studies (cf. Knorr-Cetina, 2001), and in particular, the role of ‘situated action’ in relation to interacting with computer system design (Suchman, 2007). Contemporary ‘testing’ of digital and computation infrastructures are, so Marres and Stark (2020) argue, characterised by moving into society and ‘real’ living environments, asking for a ‘sociology of testing’ capable of grappling with the multiple, social *and* technical ways in which digital, mobile and sensory media technologies are situated.

Moreover, the adoption of such distributed architecture involving apps, mobile devices and server infrastructures, which are needed to accomplish network traffic capture in use and on the move, signifies a departure from the convention of digital methods tools running on a single computer or a server. This move unavoidably increases the complexity of engineering in toolmaking practices.

Having implemented this infrastructure, we encountered the next level of resistance on the app level called ‘certificate pinning’: Some app publishers build in measures to avoid decryption even when device level certificates are in place, making it difficult for *AppTraffic* to encrypt their traffic (Bhor and Karia, 2017). Due to this ‘infrastructural resistance’ (Dieter et al., 2019: 10), data collections generated through capture sessions can feature different levels of detail, depending on the operating system and the apps used: *AppTraffic* therefore offers three capture modes: ‘Decrypted’, ‘Metadata’ and ‘Raw’. The foundation for all these capture modes is that insights from network connections are gained from so-called HTTP requests, which contain the domain name, the path, the query string, headers, cookies, the message body and so forth (Fielding and Reschke, 2014). Domain names reveal the destinations of the data in transit; and users’ data may be embedded and transferred in the paths, query strings, headers, cookies and message bodies. In the Decrypted mode, *AppTraffic* tries to offer the maximum decryption, which is also computationally most intensive: The HTTP and HTTPS requests are routed to *Mitmproxy*¹⁸ for decryption, recording data values extracted from different parts of a HTTP(S) request, namely, the URL, headers and body. In Metadata and Raw modes, the undecrypted network packets are recorded building on aforementioned capture tool *TCPDump* before being routed to the internet gateway, allowing to capture the basic information of network connections of apps with certificate pinning and on more restrictive Android devices. In Metadata mode, only the key information about the traffic, such as the destination, the time and the size of the payload, is recorded.

The sheer multiplicity of connections makes it necessary to not only get an insight into individual connections but gain different aggregate views. We therefore created basic summary analytics for each capture session that show the geographic destinations of the traffic and the data values transmitted to these destinations. Users can view and visualise such topology – as well as the intensity of app traffic over time – by importing the captured data to existing network traffic analytics tools such as *Wireshark*.¹⁹ Using *GeoIP*,²⁰ the geolocation identification feature of

Wireshark,²¹ we discovered, for example, that health app *Zepp* – an app category that is often used in conjunction with activity trackers – connected to multiple German IP addresses under the control of Chinese companies (Figure 8). That is to say, sensitive data generated by citizens of and within the European Union might be transferred to geographical locations beyond this territory, posing the question of the extent to which economic players who operate independent of the governing system of ‘European values’ (cf. Irion et al., 2021) do have access to this data, and for which purposes.²²

A key challenge we encountered is that network connection information cannot always be straightforwardly connected to the app that initiated it. *AppTraffic* allows users to see which network connections were established, but not by what app or which code within an app, adding a level of obfuscation to this research approach. As a workaround, we suggest implementing a carefully crafted approach to mobile phone usage when capturing app traffic, which includes using a dedicated research device with controlled settings, muting background noise (e.g. automated updates and app refreshes) and focussing on selected apps and in-app behaviour. This may require to re-introduce controlled and lab-like settings of use – at least to a certain extent – but also allows to single out the traffic of individual apps.

To capture the relation between practice and network connection, the next step in the development of *AppTraffic* is to bring together screen capture with network connection capture to study front-and back-end side by side. We started to conduct first trials with aforementioned apps for activity tracking (i.e. Strava, Runtastic and Fitbit), which additionally offer or support in-app, cross-app and/or cross-device activity data recording and use. For example, Fitbit allows for cross-device data retrieval (using the Fitbit device and mobile phone app) and processing through ‘body sensors’ measuring values such as one’s ‘heart rate’. We were able to access and decrypt the ‘sensed’ and processed heart rate data through network connections captured with *AppTraffic*, as it had been

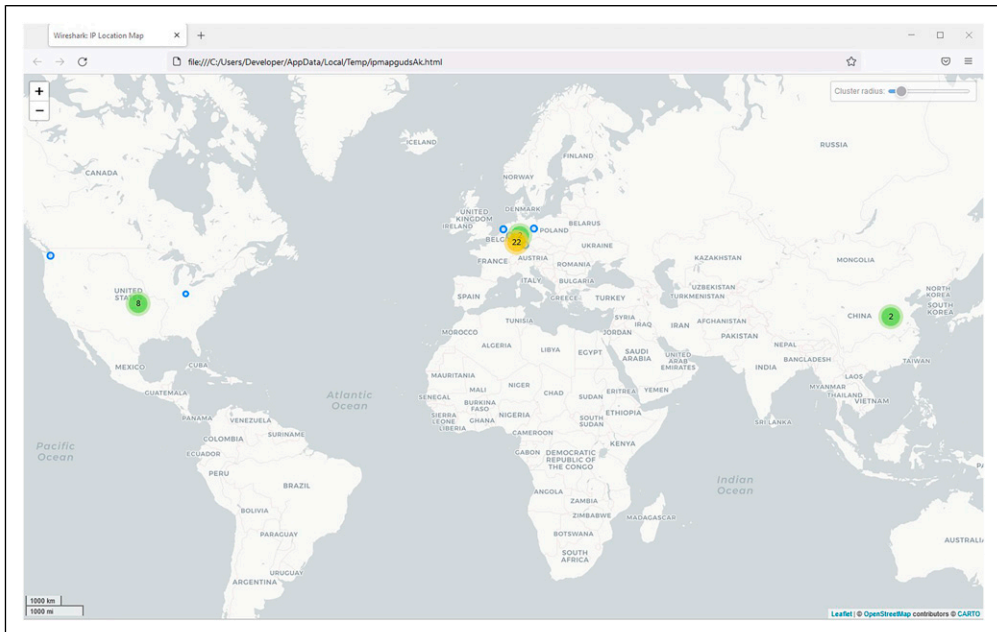


Figure 8. Wireshark GeolIP mapping based on *Zepp*’s app network traffic. Mapped are the geographical destinations of network connections made by *Zepp*.

transferred from the FitBit device to the servers that send back an interpretation of that sensor data into the app. In the FitBit app, the data is displayed in a human-readable format as heart rate time series. In that sense, [Figure 9](#) displays the back-end of this transfer and sensemaking process. This process poses not just the question, *what* but also *when* sensor data is (cf. [Salter 2022](#)), since the resulting heart rate data is the product of (partly) inaccessible calculation procedures, and needs to be computationally accomplished. Also, as ‘reading’ – including decrypting – such data is technically more complicated, it is not (yet) featured as standard functionality in *AppTraffic*, and might be demanding to implement as a generalisable approach for different tool users.²³

Making *AppTraffic* thus contributed to mobile methodologies of analysing network traffic by exploring their socio-technical and infrastructural underpinnings. As in the case of *AppInspect*, following the media was not possible in a straightforward manner. Rather, we followed the data and encountered a series of resistances. The toolmaking was realised as a CTP as it allowed researchers to experience conflicting aims when dealing with data protection, namely, when the infrastructural measures of privacy protection through modes of encryption or certificate pinning work against one’s research aims. Interdisciplinary toolmaking, realised as a cooperation between media research and computational engineering, enabled us to bring together critical and technical work in a way that contributes to understanding data flows of mobile sensory media and their socio-technical situatedness.

Discussion: Doing sensory media research?

In light of our toolmaking stories, we strive to revisit some of the guiding principles of digital methods and this methodological programme’s ‘metaphors in practice’ ([Agre, 1997a: 27–28](#)). In particular, the distinctive nature of ‘doing digital methods’ ([Rogers, 2019](#)) in (mobile) sensory media research challenges the principles of ‘following the media’ through their methods. This is related to the above highlighted complexity of these media technologies, requiring a perspective that

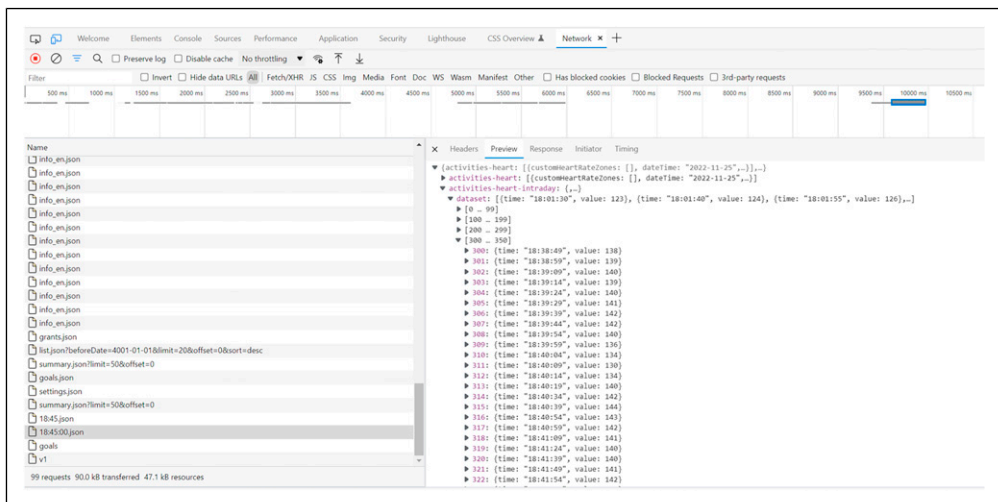


Figure 9. The captured and decrypted network traffic indicating the transfer of ‘heart rate’ data between the Fitbit app and the servers. The traffic has been captured in *AppTraffic* and opened in Google Chrome developers’ tools.

takes into account, in which ways such media are multi-situated, infrastructurally entangled and (partly) obfuscated, and through user and developer practices (re)configured. Below we propose two key considerations for exploring and imagining alternative ways of using mobile sensory media – that is, repurpose them – for research.

Attending to infrastructural entanglements

Living up to CTP in sensory media research involves putting a strong focus on the technicity of these media and their data: Developing and prototyping *AppInspect* and *AppTraffic* has shown that exploring data of sensory media, including network traffic data, requires a deep understanding of the layers of infrastructure that co-constitute this data, and the mobile sensory media user experience in general. Such an understanding of the ‘app/infrastructure stack’ (Gerlitz et al., 2019a) is critical to assess the power dynamics that have emerged among apps, platforms and other entities on the web (cf. Dieter et al., 2021). The CTP proposed in this article takes this complex layered and interconnected reality – and the involved infrastructural resistance – seriously, making an effort to unravel its entanglements to repurpose the methods of sensory media. Ultimately, this methodological outlook needs to grapple with the instability that mobile sensory media and their data feature, and therefore, their quality of being highly contingent objects of study. It requires toolmaking as continuous practices of further development and testing, maintenance of and care for the research tool and its infrastructure as well as its (future) user communities (cf. Rieder et al., 2022).

The development and prototyping of *AppInspect*, and the static modes of analysis this toolmaking process featured, has surfaced a number of research affordances and possibilities. For instance, in future projects *AppInspect* could support researchers in detecting software libraries, SDKs, and API connections that are ‘hardcoded’ into the source code of an app (or a collection of apps), including both ‘official’ and ‘unofficial’ APIs and covert developmental practices (cf. Gerlitz et al., 2019b). It is particularly important to acknowledge that mobile sensory media have facilitated ‘a more mature phase of datafication’, which is ‘best understood by examining the relationality between mobile permissions and embedded, third-party services known as SDKs’ (Pybus and Coté, 2021). Mobile in-app (micro)services have enabled infrastructural platforms like Google, Facebook and others, to extend the reach of datafication across the entire ecosystem of sensory media (cf. Blanke and Pybus, 2020). We hope that *AppInspect*, and the research and toolmaking practices it invites, contributes to the positioning of these infrastructural platforms in ecological terms. Furthermore, we envision *AppInspect* to facilitate research that aims at comparing the source code of an app (or a collection of apps) in its entirety to another app (or a collection of apps) to identify infrastructural commonalities and differences, as well as code reuse practices and template cultures (cf. Helmond et al., 2019). That is to say, CTPs with such a digital methods tool offer opportunities for critique by questioning ‘Big Tech’ platforms as ‘invasive species in the app ecosystem’ (Lai and Flensburg, 2021), emphasising the political economy of mobile communication.

The development and prototyping of *AppTraffic*, and its dynamic modes of analysis has led us to carve out research affordances of mobile sensory media’s infrastructural entanglements and the lively quality of their data ‘in the wild’ and ‘in motion’. *AppTraffic* emphasises how difficult it is to follow sensory media that appear different to separate users, in diverse locations and at distinct times of day. Network connections enable researchers to empirically study the technological and structural particularities of (mobile) sensory media such as the topology and the temporality of network connections. It is possible to explore and question the functional purpose of network connections through their contents (e.g. the actual text, images and videos transmitted, or sensor data processed). Through the toolmaking of *AppTraffic*, we realised that distinguishing *inbound* (incoming) and

outbound (outgoing) network traffic connections is useful,²⁴ because they represent different types of infrastructure and provide distinct research affordances: Inbound network connections often serve to request and load content or advertisements from remote servers, which may involve cloud storage infrastructure services and content distribution networks (CDNs). In contrast, outbound network connections often serve to communicate status information and parameters to remote servers, including analytics tools and ‘trackers’ used to localise and personalise the user experience (including targeted advertising, e.g. the possibility to offer health services based on aggregated heart rate measurements). The encryption of network traffic for privacy and security purposes hides the actual contents of these network traffic connections, which a critical technical practitioner may be interested in. As a result, the making of *AppInspect* indicated that it is vital in technical, legal and ethical sense to situate research in ways that allows for decrypting network traffic for inspection – see the contents – of the packets exchanged, such as through custom VPN connections.

Developing sensory digital methods also poses the question of the needed minimum level of technical knowledge within a research team. We needed to navigate between the aim to generate more egalitarian access to empirical media studies and the aim to invite tool users to reflect on the capacities and limits of the tools. Some of our objectives could be realised, while others such as assessing for every app which specific kinds of sensor data are collected or inferred in practice still pose methodological, technical or security challenges. Toolmaking for situated, ever-changing mobile sensory media, as our stories are supposed to demonstrate, is an ongoing, iterative, co-operative and interdisciplinary research practice. This research practice mediates between the media and the researchers’ interests rather than simply following these media, as we sometimes may need to work against them – for instance, in the case of decryption or other infrastructural resistance.

Designing sensory research situations

A second consideration is that researchers are invited – maybe even required – to design research situations, in which they ‘(re-)entangle’ themselves and play an active role in the generation of research data. Digital methods researchers are no longer able to just ‘follow’ the pre-structured and pre-formatted data sources ‘out there’. It might not always be possible or insightful for digital methods researchers to ‘disentangle’ themselves from the research devices and situations they study, since these devices and their contexts of use ‘participate’ in the research itself (cf. [Weltevrede, 2016](#)). Instead, research of sensory media benefits from studying, initiating and designing (or staging) specific research situations. This involves inventive uses of the materiality and technicity of mobile sensory media, especially in the case of dynamic – and therefore, less stable – modes of analysis.

Previously, ‘doing digital methods’ involved setting up a ‘research browser’ profile that would not track, store web cookies or identify any other personalising data, for the purpose of ‘disentangling the researcher from the results’ ([DMI Wiki, 2018](#)). In other cases, researchers set up ‘fake’ social media user profiles. With *AppTraffic*, by contrast, it is necessary to reckon with different types of user experiences, for example, by ‘staging experimental and exploratory situations’ ([Dieter et al., 2019: 2](#)). Network traffic connections and other sensory inputs, as well as dynamics of localisation and personalisation, can only be studied when they are generated through specific forms of ‘live’ usage. As a result, sensory data is always entangled with – and can only be made sense of in relation to – mundane use practices that are situated spatially and temporally, as well as online and offline.

Instead of pursuing ‘disentanglement’, researchers are encouraged to embrace the mundane quality of sensory media and their data and to actively entangle them to tease out data flows or infrastructural obfuscations. One way of devising research situations is the systematic design and

use of research personas (cf. [Bounegru et al., 2022](#)). Research personas draw from a long tradition in human-computer interaction and interaction design ([Cooper, 2004](#)) and have been put to experimental use in app research ([Bounegru et al., 2022](#); [Dieter et al., 2019](#)). They can be conveniently combined with ‘walkthroughs’ to study interface design in terms of and relation to usage ([Light et al., 2016](#) in [Dieter and Tkacz, 2020](#)). In this respect, the main challenge is to find a balance between conditions of possibility for the study of sensory media in action, in the wild and in motion, and ensuring the assessability, that is, the comparability and generalisability of research data, without ending up in a laboratory-setting again.

One way to guarantee the systematic study of sensory media usage involves creating research scenarios through a combination with ‘situational analytics’ ([Marres, 2020](#)). That is, an approach that turns to users’ practices, intentions and articulations thereof in computational settings, and takes the ‘situations’ in which the generated research data were produced and processed seriously. For instance, together with other researchers from our research centre we conducted first experiments with encoding app reviews of Strava users in relation to a major app update on 19 May 2020 that restricted large parts of the data access for free users – restrictions that have been (partially) withdrawn on 31 August 2021 ([Strava Inc, n.d.](#)). Our attempt to identify user practices’ ‘sources of normal, natural trouble’ ([Garfinkel, 1967](#): 192) in app reviews represents part of a shift from studying issues to studying practices through content and utterances on social media platforms. We are planning to use these encoded reviews in several languages (i.e. in English, German and Dutch, languages that match geographical regions where according to the Strava heatmap²⁵ large parts of the user community can be found) to develop a research protocol for staging research situations.

In that way combining digital methods with situational analytics, supports researchers in ‘sensing’ and following the full infrastructural range and implications of app usage (cf. [Ochs et al., 2021](#)) and the involved data practices. In doing so, researchers are able to both access and assess the functionality, valuation and impact of sensory media. As mentioned before, paying attention to the full picture of the ‘liveliness’ of sensor(y) data, requires next to insights into and the staging of use situations, also access to and a thorough understanding of development situations that led to, for instance, the current classification and according processing of data generated through mobile devices and their apps. The question how to organise such critical access to developers’ documentation is one central question in our future toolmaking and the documentation thereof.

Conclusion

This article presents an effort to orient digital methods to the study of mobile sensory media, that is, ‘smart’ networked devices such as smartphones and their apps, by turning to ‘critical technical practice’ (CTP), emphasising the need for technical work combined with media critique in this research endeavour. While these media are prominent in mundane practices (e.g. social interaction, navigation, activity tracking), they are also infrastructurally entangled. That is, they are dependent on ‘Big Tech’ companies, their platforms and networks, and are subject to their policies, politico-economical and technical changes, just as they are informed by privacy protection of their data flows. This position makes mobile sensory media increasingly contingent objects of study, which are not just socially, but also technically multi-situated. The infrastructural entanglements such media rely on include built-in hardware- and software-based sensors, the operating system, applications and in-app functionality and services that activate sensors in certain ways. Such sensors in turn facilitate potentially continuous, but unnoticed monitoring of our everyday (inter)activities, producing personal, (partially) sensitive data. We proposed ‘toolmaking’ as CTP, framing it as an approach and attitude to investigate this socio-technical multi-situatedness of mobile sensory media

and their data infrastructures. This perspective encourages us to explore the opportunities and limitations these media technologies and their data may hold for empirical media research. In that, we suggested toolmaking as a mode of methodological reflexivity, and the documentation of and reflection on decision-making in this process – what we call ‘toolmaking stories’ – as a productive academic genre to conjoin critical and technical practices. We presented two toolmaking stories for studying mobile sensory media through apps and their data flows: (1) We accounted for *AppInspect* and the tool’s making, drawing on static code analysis to identify sensor data access and retrieval. (2) We also presented and reflected on *AppTraffic* and the respective toolmaking, as means of tracing and studying dynamic network connections these apps enable and rely on, including their contents (e.g. text, images, and sensor data).

(Re)engaging digital methods with CTP allowed us to further develop both in the context of mobile sensory media: Thinking toolmaking for the study of such media as a specific CTP surfaces two central dimensions of being critical with and through digital methods. On the one hand, toolmaking highlights infrastructural entanglements of sensory media: It opens up avenues to develop a critical stance with respect to the phenomenon of study, the role of global infrastructure providers, code politics, data governance or security settings of computational systems. Tools like *AppInspect* and *AppTraffic*, and their making stories, allow researchers to reveal and inquire into the workings of mobile sensory media, including the many layers and connections that are invisible or purposefully hidden such as network traffic activated in the background or when apps are closed. Put differently, toolmaking contributes to data and infrastructure literacy, disclosing the ways in which computational systems and their data are encrypted and otherwise obfuscated. Such technical mechanisms may serve to protect users’ privacy, but also limit researchers’ ability to explore and define the research affordances of digital media, and thus, jeopardise researchers’ possibility to monitor ‘Big Tech’ companies, their platforms and services, in order to hold these parties responsible.

On the other hand, the notion of CTP stimulates reflection on researchers’ assumptions and practices of doing digital methods that they are confronted with when attempting to carve out the research affordances of mobile sensory media. Due to sensory media’s infrastructural entanglements and obfuscation, their study challenges key ideas and approaches of digital methods, that is, concepts and approaches like following, repurposing, working with predefined digital objects, and disentangling the researcher from the method. We demonstrated that toolmaking becomes imperative to even access (mobile) sensory media for investigation. Moreover, the study of such media encourages – and maybe even asks for – the design of research situations, in which empirically invested researchers are invited to (re)entangle themselves in the generation and sensemaking of lively research data, as an opportunity to explore mobile sensory media usage in the wild.

In this line of argumentation, ‘sensing’ emerges as a broader phenomenon, involving network traffic connections that ‘sense’ and ‘situate’ the user experience in certain ways – not unlike the search engine that ‘senses’ the user’s or browser’s location to set the country code and default language, which in turn influences the results returned. Thus, sensory media studies need to unravel and identify sensing mechanisms that are built into the digital media technologies we increasingly use in everyday life, and consider what exactly is sensed, when, how and why, including at which locations and through which activities.

Strikingly, many networked mobile devices are considered ‘personal’, when in fact they are ‘chatty and promiscuous’ (Chun, 2016). Notably, sensory media extend beyond what is considered to be personal or intimate; they are also networked and embedded in and mobilised through vehicles, homes, factories, hospitals, traffic situations, cities and more. We argue that in each of these contexts and settings, it is vital to cultivate and embrace CTPs, not just to identify the

particularities and methods of these media, but also to develop, maintain, and support others in developing a literacy for the intricate data infrastructures associated with them.

Notes

1. This conception of media relates to a practice-theoretical understanding, in which media are always the product of situated practices and the socio-materialities that these practices are embedded in and (re)configure (cf. Schüttpelz et al., 2021).
2. Because of their importance to digital methods research practices, the continuous policy and technical updates to search engines, platforms and APIs (and sometimes their deprecation), have already altered and broken many digital methods research tools, including the Netvizz application for Facebook (Rieder, 2013). See also: <https://thepoliticsofsystems.net/2015/01/the-end-of-netvizz/>.
3. For example, https://developer.android.com/guide/topics/sensors/sensors_motion.
4. Toolmaking stories, then, are comparable to what has been coined ‘data stories’ (Mosconi et al., 2022), for example, in ethnographically informed research practice. Yet, in toolmaking stories the focus is less on the research material and more on the description of and reflection on the digital methodologies carved out and devised to access, collect and make sense of this material.
5. For more information see <http://appstudies.org/>.
6. See <https://developer.android.com/training/permissions/requesting> and <https://developer.apple.com/design/human-interface-guidelines/ios/app-architecture/accessing-user-data/>.
7. Cf. <https://developer.android.com/reference/android/hardware/SensorEventListener>.
8. Creating an Android app using Android Studio – the official Android app development environment – automatically generates a Manifest file with all the elements required for apps’ operations on an Android device.
9. Cf. https://developer.android.com/reference/android/provider/MediaStore#ACTION_IMAGE_CAPTURE.
10. See <https://developer.android.com/reference/android/hardware/SensorManager>.
11. This research endeavour is closely connected to the PhD project of Daniela van Geenen, also in media studies.
12. From resources such as <https://www.apkmirror.com/> and <https://apkpure.com/app>.
13. Not only the number of dangerous permissions increased, but also the number of permissions in general almost doubled, which also points to increased platform governance, including in reaction to aforementioned data scandals.
14. See also <https://www.wireshark.org/>.
15. See also <https://www.tcpdump.org/>.
16. Less common, but also possible when legally allowed and ethically responsible, is to request the network data of mobile devices directly from telecommunications operators (cf. Ghahramani et al., 2020; Jiang et al., 2019).
17. It usually works for iOS and older versions of Android, but not for devices using versions 7 and above (Brubaker, 2016).
18. See <https://mitmproxy.org/>.
19. The documentation surrounding *AppTraffic* (on GitHub) guides the user through how to access and analyse the captured data, see: https://github.com/jason-chao/workshops/blob/main/2021/Feb-SMART2021/blackbox_of_mobile_apps_traffic.md. Since *Wireshark* already offers thorough opportunities to view and visualise data, this part has not been built into *AppTraffic*, but both tools can be combined and concatenated in users’ analysis practices.
20. See <https://wiki.wireshark.org/HowToUseGeoIP>.

21. For *Wireshark* to identify and assign geolocations, the *MaxMind GeoLite2* database needs to be imported into the tool. See <https://dev.maxmind.com/geoip/geo-lite2-free-geolocation-data>.
22. While legal measures (e.g. agreements for cloud data storage) and technical means (e.g. encryption) might be in place, it is a challenge to not just trace the involved (third) parties, but also enquire into the kinds of regulations these actors comply with. Moreover, as our experiments with the Fitbit activity tracker illustrate, which we discuss below, computational forms of data protection such as encryption can be implemented in more or less strict and technically (im)permeable ways.
23. What might complicate the toolmaking here even more is that, at a certain point, such sensor data might also be subject to further restrictions due to progressing encryption practices.
24. These can be, for instance, accessed and distinguished through a combination of *AppTraffic* for capturing network connections and *Wireshark* to view the captured data.
25. Available at: <https://www.strava.com/heatmap#3.30/-0.28602/47.96550/hot/all>.

Acknowledgments

‘We would like to thank participants of pilot studies in the area of app studies conducted during the 2018–2019 Digital Methods Summer and Winter Schools at the University of Amsterdam. Special thanks to Anne Helmond, Esther Weltevrede, Michael Dieter, Nathaniel Tkacz, and Noortje Marres for their collaboration. We also want to thank Loup Cellard, Robin de Mourat, Danny Lämmerhirt, Stefan Laser for their collaboration on experiments during the 2021 Mixed Methods Summer School at the University of Siegen. Moreover, we are grateful for the lively exchanges we had with (former) colleagues from the CRC “Media of Cooperation” and other participants during the “Sensing the Sensor” Workshop in November 2022 discussing our empirical cases. Lastly, we want to express our gratitude to the journal editors, special issue guest editors, the special issue contributors and (anonymous) reviewers for their thoughtful and thorough commentary on earlier versions of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Deutsche Forschungsgemeinschaft (Project number 262513311 (SFB 1187: ‘Media of Cooperation’)).

References

- Agre PE (1997a) *Computation and Human Experience*. Cambridge: Cambridge University Press. DOI: [10.1017/CBO9780511571169](https://doi.org/10.1017/CBO9780511571169)
- Agre PE (1997b) Toward a critical technical practice: lessons learned in trying to reform AI. In: GC Bowker, SL Star, L Gasser, et al. (eds) *Social Science, Technical Systems, and Cooperative Work: Beyond the Great Divide. Computers, Cognition, and Work*. Mahwah, NJ: Lawrence Erlbaum Associates, 131–157.
- Andrejevic M and Burdon M (2015) Defining the sensor society. *Television & New Media* 16(1): 19–36. DOI: [10.1177/1527476414541552](https://doi.org/10.1177/1527476414541552)
- Android Developers (2022a) <permission>. Available at: <https://developer.android.com/guide/topics/manifest/permission-element> (accessed 17 April 2023).
- Android Developers (2022b) Security with network protocols. Available at: <https://developer.android.com/training/articles/security-ssl> (accessed 17 April 2023).
- Android Developers (2023) Minimize your permission requests. Available at: <https://developer.android.com/training/permissions/evaluating> (accessed 17 April 2023).
- Apple Support (2022) Trust manually installed certificate profiles in iOS and iPadOS. Available at: <https://support.apple.com/en-us/HT204477> (accessed 17 April 2023).

- Babich N (2017) Wizard design pattern. In: *Medium*. Available at: <https://uxplanet.org/wizard-design-pattern-8c86e14f2a38>
- Bhor M and Karia D (2017) Certificate pinning for android applications. In: 2017 International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 19–20 January, 2017, pp. 1–4. DOI: [10.1109/ICISC.2017.8068748](https://doi.org/10.1109/ICISC.2017.8068748)
- Binns R, Lyngs U, Van Kleek M, et al. (2018) Third party tracking in the mobile ecosystem. In: Proceedings of the 10th ACM Conference on Web Science, New York, NY, USA, 15 May, 2018, Association for Computing Machinery, 23–31. WebSci '18. DOI: [10.1145/3201064.3201089](https://doi.org/10.1145/3201064.3201089)
- Blanke T and Pybus J (2020) The material conditions of platforms: monopolization through decentralization. *Social Media + Society* 6(4): 1–13. DOI: [10.1177/2056305120971632](https://doi.org/10.1177/2056305120971632)
- Bounegru L, Devries M, and Weltevrede E (2022) The research persona method: figuring and reconfiguring personalised information flows. In: C Lury, W Viney, and S Wark (eds) *Figure: Concept and Method*. Singapore: Springer Nature, 77–104. DOI: [10.1007/978-981-19-2476-7_5](https://doi.org/10.1007/978-981-19-2476-7_5)
- Brubaker C (2016) Changes to trusted certificate authorities in android nougat. In: *Android Developers Blog*. Available at: <https://android-developers.googleblog.com/2016/07/changes-to-trusted-certificate.html>
- Bunz M and Meikle G (2017) *The Internet of Things*. Digital media and society. Cambridge, UK: Polity Press.
- Caliandro A (2021) Repurposing digital methods in a post-API research environment: methodological and ethical implications. *Italian Sociological Review* 11(4S): 225–242. DOI: [10.13136/isr.v11i4S.433](https://doi.org/10.13136/isr.v11i4S.433)
- Chao, J (2023a) AppInspect (Frontend Component) v0.0.1 beta. Zenodo, 7 March. DOI: [10.5281/ZENODO.7705032](https://doi.org/10.5281/ZENODO.7705032)
- Chao J (2023b). jason-chao/AppTraffic: AppTraffic v0.0.1 beta. Zenodo. [10.5281/zenodo.7704963](https://doi.org/10.5281/zenodo.7704963)
- Chao J (2023c). jason-chao/appinspect-services: AppInspect (Services Component) v0.0.1 beta (v0.0.1beta). Zenodo. DOI: [10.5281/zenodo.7705028](https://doi.org/10.5281/zenodo.7705028)
- Chao J (2023d) jason-chao/AppTraffic: AppTraffic v0.0.1 beta. Zenodo. DOI: [10.5281/zenodo.7704963](https://doi.org/10.5281/zenodo.7704963)
- Chen K, Zhang D, Yao L, et al. (2021) Deep learning for sensor-based human activity recognition: overview, challenges, and opportunities. *ACM Computing Surveys* 54(4): 77:1–77:40. DOI: [10.1145/3447744](https://doi.org/10.1145/3447744)
- Chun WHK (2016) *Updating to Remain the Same: Habitual New Media*. Cambridge: MIT Press.
- Cooper A (2004) *Inmates Are Running the Asylum, the: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity*. Indianapolis, IN: Sams - Pearson Education.
- Cousineau LS, Oakes H, and Johnson CW (2019) Appnography: modifying ethnography for app-based culture. In: Parry DC, Johnson CW, and Fullagar S (eds). *Digital Dilemmas: Transforming Gender Identities and Power Relations in Everyday Life*. Cham: Palgrave Macmillan, 95–117. Available at: https://doi.org/10.1007/978-3-319-95300-7_5.
- Crawford K (2021) *Atlas of AI: Power, Politics, and the Planetary Costs of Artificial Intelligence*. New Haven: Yale University Press.
- Dieter M and Tkacz N (2020) The patterning of finance/security: a designerly walkthrough of challenger banking apps. *Computational Culture* 2020 (20th January), 1–38. Available at: <https://computationalculture.net/the-patterning-of-finance-security/>
- Dieter M, Gerlitz C, Helmond A, et al. (2019) Multi-situated app studies: methods and propositions. *Social Media + Society* 5(2): 1–15. DOI: [10.1177/2056305119846486](https://doi.org/10.1177/2056305119846486)
- Dieter M, Helmond A, Tkacz N, et al. (2021) Pandemic platform governance: mapping the global ecosystem of COVID-19 response apps. *Internet Policy Review* 10(3): 1–28. DOI: [10.14763/2021.3.1568](https://doi.org/10.14763/2021.3.1568)
- DMI Wiki (2018) Things internet researchers should know about search engines. Available at: <https://wiki.digitalmethods.net/Dmi/ThingsInternetResearchersShouldKnowAboutGoogle>
- Ferretti L, Wymant C, Kendall M, et al. (2020) Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing. *Science* 368(6491): eabb6936. DOI: [10.1126/science.abb6936](https://doi.org/10.1126/science.abb6936)

- Fielding R and Reschke JF (2014) Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing. Internet Engineering Task Force (IETF). Available at: <https://httpwg.org/specs/rfc7230.html>
- Garfinkel H (1967) *Studies in Ethnomethodology*. Englewood Cliffs: Prentice-Hall.
- Gekker A and Hind S (2020) Infrastructural surveillance. *New Media & Society* 22(8): 1414–1436. DOI: [10.1177/1461444819879426](https://doi.org/10.1177/1461444819879426)
- Gerlitz C and Helmond A (2013) The like economy: social buttons and the data-intensive web. *New Media & Society* 15(8): 1348–1365. DOI: [10.1177/1461444812472322](https://doi.org/10.1177/1461444812472322)
- Gerlitz C, Helmond A, Nieborg DB, et al. (2019a) Apps and infrastructures – a research agenda. *Computational Culture* 7 (21st October 2019), 1–30. Available at: <https://computationalculture.net/apps-and-infrastructures-a-research-agenda/>
- Gerlitz C, Helmond A, van der Vlist FN, et al. (2019b) Regramming the platform: infrastructural relations between apps and social media. *Computational Culture* 7 (21st October 2019), 1–44. Available at: <https://computationalculture.net/regramming-the-platform/>
- Ghahramani M, Zhou M, and Wang G (2020) Urban sensing based on mobile phone data: approaches, applications, and challenges. *IEEE/CAA Journal of Automatica Sinica* 7(3): 627–637. DOI: [10.1109/JAS.2020.1003120](https://doi.org/10.1109/JAS.2020.1003120)
- Gray J, Gerlitz C, and Bounegru L (2018) Data infrastructure literacy. *Big Data & Society* 5(2): 1–13. DOI: [10.1177/2053951718786316](https://doi.org/10.1177/2053951718786316)
- Helmond A, van der Vlist FN, Weltevrede E, et al. (2018) Mapping data-intensive app infrastructures. In: *WinterSchool2018*. Available at: <https://wiki.digitalmethods.net/Dmi/WinterSchool2018MappingDataIntensiveAppInfrastructures>
- Helmond A, van der Vlist FN, Weltevrede E, et al. (2019) App stores and their bias: repurposing ‘app relatedness’. In: *SummerSchool2018*. Available at: <https://wiki.digitalmethods.net/Dmi/SummerSchool2018AppStoresBias>
- Irion K, Burri M, Kolk A, et al. (2021) Governing “European values” inside data flows: interdisciplinary perspectives. *Internet Policy Review* 10(3): 1–14, Available at: <https://policyreview.info/european-values>
- Jacomy M, Venturini T, Heymann S, et al. (2014) ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. *PLoS One* 9(6): e98679. DOI: [10.1371/journal.pone.0098679](https://doi.org/10.1371/journal.pone.0098679)
- Jiang Y, Du X, and Jin T (2019) Using combined network information to predict mobile application usage. *Physica A: Statistical Mechanics and its Applications* 515: 430–439. DOI: [10.1016/j.physa.2018.09.135](https://doi.org/10.1016/j.physa.2018.09.135)
- Kanderske M and Thielmann T (2019) *Simultaneous Localization and Mapping and the Situateness of a New Generation of Geomedial Technologies: Communication and the Public*. DOI: [10.1177/2057047319851208](https://doi.org/10.1177/2057047319851208)
- Klimburg-Witjes N, Poehhacker N, and Bowker GC (eds) (2021) *Sensing In/Security: Sensors as Transnational Security Infrastructures*. Manchester: Mattering Press. DOI: [10.28938/9781912729111](https://doi.org/10.28938/9781912729111)
- Knorr Cetina K (2001) Laboratory studies: historical perspectives. In: NJ Smelser and PB Baltes (eds) *International Encyclopedia of the Social & Behavioral Sciences*. Oxford: Pergamon, 8232–8238. DOI: [10.1016/B0-08-043076-7/03175-2](https://doi.org/10.1016/B0-08-043076-7/03175-2)
- Lai SS and Flensburg S (2021) Invasive species of the app ecosystem: exploring the political economy of mobile communication. *International Journal of Communication* 15: 2301. Available at: <https://ijoc.org/index.php/ijoc/article/view/16906>
- Light B, Burgess J, and Duguay S (2016) The walkthrough method: an approach to the study of apps. *New Media & Society* 20(3): 881–900. DOI: [10.1177/1461444816675438](https://doi.org/10.1177/1461444816675438)
- Lupton D (2016) *The Quantified Self*. Cambridge: Polity Press.
- Mackenzie A (2010) *Wirelessness: Radical Empiricism in Network Cultures*. Cambridge, Mass: MIT Press.
- Marres N (2020) For a situational analytics: an interpretative methodology for the study of situations in computational settings. *Big Data & Society* 7(2): 1–9. DOI: [10.1177/2053951720949571](https://doi.org/10.1177/2053951720949571)

- Marres N and Stark D (2020) Put to the test: for a new sociology of testing. *British Journal of Sociology* 71(3): 423–443. DOI: [10.1111/1468-4446.12746](https://doi.org/10.1111/1468-4446.12746)
- Marres N and Weltevrede E (2013) Scraping the social? *Journal of Cultural Economy* 6(3): 313–335, Routledge. DOI: [10.1080/17530350.2013.772070](https://doi.org/10.1080/17530350.2013.772070)
- Mosconi G, Karasti H, Randall D, et al. (2022) Designing a data story: an innovative approach for the selective care of qualitative and ethnographic data. In: M Burkhardt, D van Geenen, C Gerlitz, et al. (eds) *Interrogating Datafication: Towards a Praxeology of Data*. Bielefeld: Transcript, 207–230.
- Nafus D (2018) Working ethnographically with sensor data. In: H Knox and D Nafus (eds) *Ethnography for a Data-Saturated World*. Manchester: Manchester University Press. DOI: [10.7765/9781526127600.00019](https://doi.org/10.7765/9781526127600.00019)
- Ochs C, Büttner B, and Lamla J (2021) Trading social visibility for economic amenability: data-based value translation on a “health and fitness platform”. *Science, Technology & Human Values* 46(3): 480–506. DOI: [10.1177/0162243920928138](https://doi.org/10.1177/0162243920928138)
- Perriam J, Birkbak A, and Freeman A (2020) Digital methods in a post-API environment. *International Journal of Social Research Methodology* 23(3): 277–290. DOI: [10.1080/13645579.2019.1682840](https://doi.org/10.1080/13645579.2019.1682840)
- Pink S, Sumartojo S, Lupton D, et al. (2017) Mundane data: the routines, contingencies and accomplishments of digital living. *Big Data & Society* 4(1): 1–12. DOI: [10.1177/2053951717700924](https://doi.org/10.1177/2053951717700924)
- Pridmore J, Zimmer M, Vitak J, et al. (2019) Intelligent personal assistants and the intercultural negotiations of dataveillance in platformed households. *Surveillance and Society* 17(1/2): 125–131. DOI: [10.24908/ss.v17i1/2.12936](https://doi.org/10.24908/ss.v17i1/2.12936)
- Professional American Service System Corp LLC. (2020) Network Connections. Available at: <https://www.network-connections.com/> (accessed 13 January 2024).
- Pybus J and Coté M (2021) Did you give permission? Datafication in the mobile ecosystem. *Information, Communication & Society* 25(11), 1650. DOI: [10.1080/1369118X.2021.1877771](https://doi.org/10.1080/1369118X.2021.1877771)
- Rettberg JW (2020) Situated data analysis: a new method for analysing encoded power relationships in social media platforms and apps. *Humanities and Social Sciences Communications* 7(1): 5. DOI: [10.1057/s41599-020-0495-3](https://doi.org/10.1057/s41599-020-0495-3)
- Rieder B (2013) Studying Facebook via data extraction: the netvizz application. In: Proceedings of the 5th Annual ACM Web Science Conference, New York, NY, USA, 2013, pp. 346–355. WebSci '13. ACM. DOI: [10.1145/2464464.2464475](https://doi.org/10.1145/2464464.2464475)
- Rieder B, Peeters S, and Borra E (2022) From tool to tool-making: reflections on authorship in social media research software. *Convergence*. DOI: [10.1177/13548565221127094](https://doi.org/10.1177/13548565221127094)
- Rieder B and Röhle T (2017) Digital methods: from challenges to bildung. In: Schäfer MT and van Es K (eds) *The Datafied Society: Studying Culture through Data*. Amsterdam: Amsterdam University Press, 109–124. Available at: <https://www.jstor.org/stable/j.ctt1v2xsqn>
- Rogers R (2013) *Digital Methods*. Cambridge: MIT Press.
- Rogers R (2019) *Doing Digital Methods*. Thousand Oaks, CA: Sage Publications.
- Ruppert E, Law J, and Savage M (2013) Reassembling social science methods: the challenge of digital devices. *Theory, Culture & Society* 30(4): 22–46. DOI: [10.1177/0263276413484941](https://doi.org/10.1177/0263276413484941)
- Salter C (2022) *Sensing Machines: How Sensors Shape Our Everyday Life*. Cambridge, Massachusetts: The MIT Press.
- Savage M and Burrows R (2007) The coming crisis of empirical sociology. *Sociology* 41(5): 885–899. DOI: [10.1177/0038038507080443](https://doi.org/10.1177/0038038507080443)
- Schüttpelz E, Bergermann U, Dommann M, et al. (2021) *Connect and Divide. The Practice Turn in Media Studies*. Berlin: Diaphanes Verlag.
- Strava Inc. (n.d.) What’s New. Available at: <https://www.strava.com/whats-new> (accessed 25 October 2021).
- Suchman LA (2007) *Human-Machine Reconfigurations: Plans and Situated Actions*. 2nd edition Cambridge: Cambridge University Press.
- Tickoo O and Iyer R (2017) *Making Sense of Sensors: Implementing a Knowledge Pipeline*. USA: Apress.

- van der Vlist FN, Helmond A, Burkhardt M, et al. (2021) *The Technicity of Platform Governance: Structure and Evolution of Facebook's APIs*. Working paper series/SFB 1187 Medien der Kooperation. Marburg, Siegen: Universitätsbibliothek Siegen. DOI: [10.25819/UBSI/9951](https://doi.org/10.25819/UBSI/9951)
- van Geenen D (2020) Critical affordance analysis for digital methods: the case of Gephi. Burkhardt M, Shnayien M, and Grashöfer K (eds) *Explorations in Digital Cultures*. Lüneburg: Meson Press. Available at: DOI: [10.14619/1716](https://doi.org/10.14619/1716)
- Venturini T and Rogers R (2019) “API-based research” or how can digital sociology and journalism studies learn from the Facebook and Cambridge analytica data breach. *Digital Journalism* 7(4): 532–540. DOI: [10.1080/21670811.2019.1591927](https://doi.org/10.1080/21670811.2019.1591927)
- Weltevrede EJT (2016) *Repurposing Digital Methods: The Research Affordances of Platforms and Engines*. Dissertation. Amsterdam: University of Amsterdam. Available at: <https://hdl.handle.net/11245/1.505660>
- Weltevrede E and Jansen F (2019) Infrastructures of intimate data: mapping the inbound and outbound data flows of dating apps. *Computational Culture 2019* (21st October), 1-39. Available at: <https://computationalculture.net/infrastructures-of-intimate-data-mapping-the-inbound-and-outbound-data-flows-of-dating-apps/>
- Wieringa M, van Geenen D, van Es K, et al. (2019) The Fieldnotes Plugin: making network visualisation in Gephi accountable. In: Daly A, Devitt SK, Mann M, et al. (eds) *Good Data. Theory on Demand*. Amsterdam: Institute of Network Cultures, 277–296.
- Wilken R, Burgess J, and Albury K (2019) Dating apps and data markets: a political economy of communication approach. *Computational Culture 2019* (21st October), 1-36. Available at: <https://computationalculture.net/dating-apps-and-data-markets-a-political-economy-of-communication-approach/>
- Wu T, Liu J, Xu Z, et al. (2016) Light-weight, inter-procedural and callback-aware resource leak detection for android apps. *IEEE Transactions on Software Engineering* 42(11): 1054–1076. DOI: [10.1109/TSE.2016.2547385](https://doi.org/10.1109/TSE.2016.2547385)
- Zhang H, Luo S, Zhang Y, et al. (2019) An efficient android malware detection system based on method-level behavioral semantic analysis. *IEEE Access* 7: 69246–69256. DOI: [10.1109/ACCESS.2019.2919796](https://doi.org/10.1109/ACCESS.2019.2919796)
- Zittrain J (2008) *The Future of the Internet—And How to Stop it*. New Haven, CT: Yale University Press.