App support ecologies

An empirical investigation of app–platform relations

Prof. Dr. Carolin Gerlitz [1,2], **Fernando N. van der Vlist** [1,2], **Dr. Anne Helmond** [2], **Dr. Esther Weltevrede** [2]

Author affiliations

[1] *(DFG Collaborative Research Centre 1187 'Media of Cooperation',) University of Siegen, Germany*

[2] *(Digital Methods Initiative,) University of Amsterdam, the Netherlands*

Keywords

mobile apps, digital methods, Google Play, platforms, social media

Download or share

`bit.ly/app-support-ecologies`

In: *Infrastructures of Publics – Publics of Infrastructures*, First Annual Conference 2016 of the DFG Collaborative Research Centre 1187 'Media of Cooperation,' Artur-Woll-Haus, University of Siegen, Germany, December 8–10.

Corresponding author: Fernando N. van der Vlist (`fernando.vandervlist@uni-siegen.de`)

Introduction

Interfacing apps and platform studies

In recent years, apps have been addressed from a number of methodological and disciplinary perspectives. The majority of contributions have focused on apps from a single medium perspective, attending to their interfaces and affordances, their affective and sensory capacities, or have explored questions of political economy and privacy in relation to specific mobile apps.

However, a growing number of apps are not designed as self-sufficient or standalone software objects, but are embedded in, built on top of or in relation to social media platforms and their databases. Social media platforms, we argue, create the socio-technical conditions for related mobile apps. And through them, these platforms are continuously enhanced, expanded, reinterpreted, and reduced or simplified through mobile apps that take up and recombine platform data and functionalities.

This project sets out to advance the study of mobile apps at the intersection with platform studies and explores what both fields of study may learn from each other. A novel empirical methodology is developed to explore the intricate relations between mobile apps and social media platforms. Our findings suggest to think of apps as relational software entities, simultaneously situated and distributed. Apps exist as part of wider ecologies made up of programmable infrastructures and controlled data flows.

Platforms define and set the conditions within which developers of apps can participate in reinterpreting platform data and platform features and repurpose them for their own objectives, most notably through the provision of APIs, their rules and regulations, and documentation. In this study we focus specifically on apps built 'on top of' or in relation to platforms, addressing both wider and narrower publics, and creating distinct interfaces between developers and users to jointly engage in reinterpreting and extending platform-specific affordances.

Platforms are increasingly reliant on developers since envisioned use cases, as well as meaning and value production, are not fixed but open to an 'interpretative flexibility' (Bijker & Pinch, 1987). As a result, platforms have effectively started to (partially) outsource exploration of, and experimentation with, their own 'interpretative flexibility' to developers testing new ideas, novel recombinations with data, and innovative use cases, extending interpretations of what a platform is for (e.g., Bucher, 2013).

Making engagement with platforms visible

To investigate this intricate, dynamic relation between mobile apps and platforms, we are interested in the following two leading research questions:

1. First, how do developers engage with and / or realise the 'interpretative flexibility' of social media platforms?
2. Second, how do platforms negotiate and respond to such continuous enhancements, expansions, reinterpretations, and reductions?

Method

Retrieving and analysing ways of relating to platforms

In order to investigate these questions, we have devised a novel empirical method for mapping diverse app–platform relations, which we describe in terms of platform and app support ecologies. We have done so by first compiling a source list of social media platforms that are among the most popular worldwide. We have then 'repurposed' (Rogers, 2013) the analytical capacities of Google Play – one of the major app stores – to find apps built 'on top of' or in relation to these source platforms, and to trace the emergent relations between these apps as they emerge within this native environment (cf. Alhar et al., 2016). The methodological protocol for this process is as follows:

1. First, creating a selection or source list of apps, platforms, or games. For present purposes, we start with an 'expert list' – one compiled by an authority on the subject matter – of the 'most popular apps in the world' based on the estimated number of global app downloads in app stores, according to Statista, one of the leading online statistics companies. As of May 2016, their list includes the following social media platforms or apps: Facebook (ranked third), Snapchat (fourth) Instagram (fifth), and Twitter (ninth) (Richter, 2016).
2. Second, querying Google Play for each source app separately (i.e., [Facebook], [Snapchat], [Instagram], and [Twitter]). We turn specifically to app stores, since they provide a native environment within which users and developers most often encounter apps (Figure 1a).

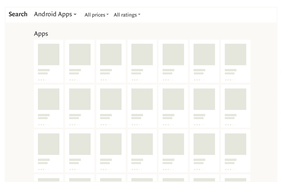

Figure 1a. Querying Android Apps on Google Play
Typical interface for browsing search query results for Android Apps on Google Play, within any price category, and with any rating (play.google.com/store/search).

3. Third, retrieving the URLs for each of these queries, for all matching results until reaching the cut-off point (in our case the 100th result, or before clicking 'Show More').
4. Fourth, stripping unique identifiers (IDs) of these apps from their URLs (e.g., Instagram's reads `com.instagram.android`).
5. Fifth, feeding app IDs into a custom-built tool named Google Play Similar Apps (Digital Methods Initiative, 2015). This scraper device extracts app details for a given list of apps as well as details for apps listed as 'Similar' on Google Play (Figure 1b).

Figure 1b. Details for Android Apps on Google Play
Typical interface for Android Apps on Google Play. All apps on Google Play have a public-facing details page, and provide a list of 'Similar' apps.

6. Finally, for analysis of the scraped results we devised a semi-automated method for emergent categorisation informed by first probing keyword resonance and then ranking searches by relevance scores, and making use of app details such as titles and text descriptions provided by the actors themselves. More specifically, we employ descriptions of apps to characterise forms of support per platform – the ways in which they enhance, expand, reinterpret, or reduce platform data and features.

Findings

Summary of findings

In our sample of four platforms, we have found both shared and very specific kinds of support apps. Most significantly, mobile apps mainly take up platform data and functionalities, whilst developers seem to pay less attention to user-generated content.

We can identify at least *three kinds of shared support apps*:

1. First, a significant portion of apps focuses on *popularity growth and strategic engagement* with platforms and their reputation systems.
2. Second, there are many apps focusing on *enhancing existing platform functionalities*, for instance by enabling users to create and edit content, enhance selfies, and offering alternative clients.
3. Third, many apps claim to *add (novel) functionalities to platforms*, which are typically not supported by the platform itself.

Visualising ways of relating to platforms

To further investigate each of the support app ecologies in relation to the four selected major social media platforms, we have produced two kinds of visualisations:

1. First, a set of so-called *sunburst diagrams (radial treemaps)*, illustrating the kinds of apps that constitute the support ecology for each platform, and their hierarchical distribution (Figures 2a–2d). Higher-ranking partitions are shown towards the centre of the diagram (root categories), and the more fine-grained partitions extend outwards into the peripheries (branching or nested categories). The centre counts refer to the number of relevant support apps found per platform. The proportion of these support apps to other 'Similar' results is shown on the lower-left side.
2. Second, a set of *network diagrams*, visualising clusters of support apps and the relations that exist between them, derived from Google Play's own mechanisms used to determine 'Similar' apps (Figures 3a–3d). Used in combination with the sunburst diagrams, we are able to delineate and locate support clusters for each of the platforms with a high degree of accuracy (e.g., in contrast to using a calculated modularity measure based on the relative density of connections).

At first glance, both Facebook's and Twitter's app ecologies have a similar profile in terms of their support ecologies: both platforms are characterised first and foremost by apps that enhance existing functionalities, most notably by offering alternative clients, apps for sharing or downloading content, image editing and camera-based apps, and providers of existing visual or textual content. However, what distinguishes these two platforms is that there are more alternative clients and content managers for Twitter as well as more apps for strategic engagement, follower and audience growth, and more apps for searching and discovering new content. Facebook, on the other hand, has more profile and activity monitoring apps. Instagram and Snapchat also show similarities as both platforms assemble a large array of image editors and apps for popularity growth. Using network diagrams, we can further explore and distinguish among characteristics for each support ecology.

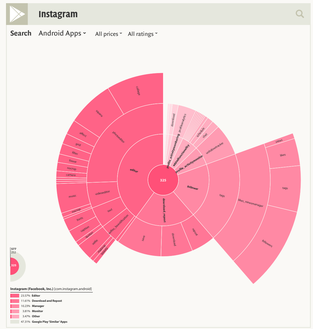

Figure 2c. Sunburst diagram for [Instagram]
Sunburst diagram for Instagram's app support ecology. 325 out of 577 results qualified as relevant, and 23.57% are 'Editor' apps (e.g., photo and video editors, selfie-making apps, music videos, and 'no-crop' square photos). 11.61% are 'Download and Repost' apps (e.g., user and hashtag analytics, popularity growth, and profile and activity monitors).

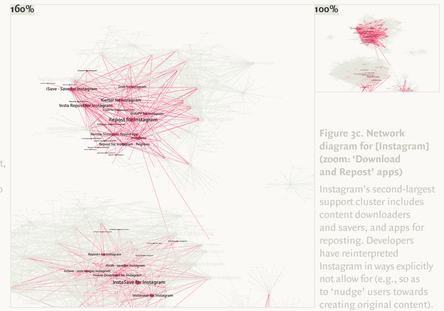

Figure 3c. Network diagram for [Instagram] (zoom: 'Download and Repost' apps)
Instagram's second-largest support cluster includes content downloaders and savers, and apps for reposting. Developers have reinterpreted Instagram in ways explicitly not allow for (e.g., so as to 'nudge' users towards creating original content).

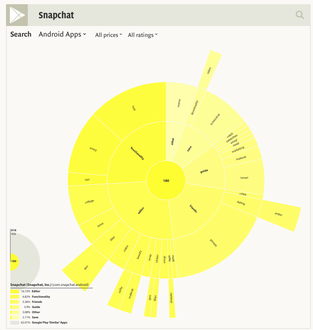

Figure 2d. Sunburst diagram for [Snapchat]
Sunburst diagram for Snapchat's app support ecology. 180 out of 616 results qualified as relevant, and 16.72% of these are 'Editor' apps, (e.g., lenses, formats, beautification, and selfie-enhancement apps). 6.82% are 'Functionality' apps (e.g., battery life savers, app locks, popularity growth, and dating-related apps).

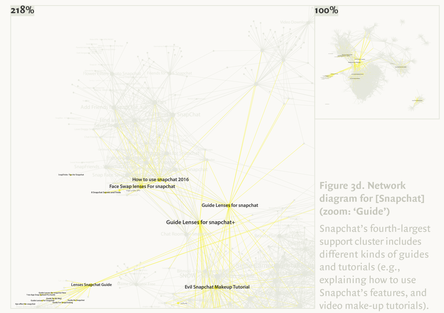

Figure 3d. Network diagram for [Snapchat] (zoom: 'Guide')
Snapchat's fourth-largest support cluster includes different kinds of guides and tutorials (e.g., explaining how to use Snapchat's features, and video make-up tutorials).

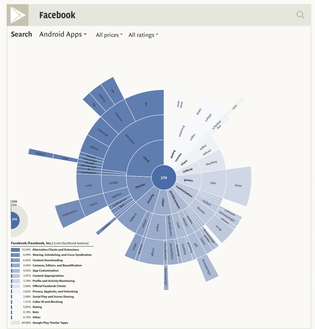

Figure 2a. Sunburst diagram for [Facebook]
Sunburst diagram for Facebook's app support ecology. 270 out of 529 results qualified as relevant, and 10.59% of these are 'Alternative Clients and Extensions'. 6.99% are 'Sharing, Scheduling, and Cross-Syndication' apps.

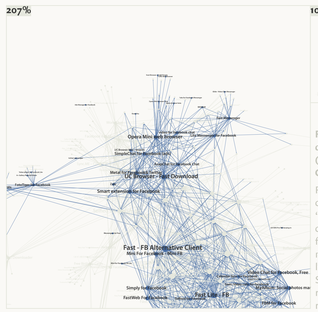

Figure 3a. Network diagram for [Facebook] (zoom: 'Alternative Clients and Extensions')
Facebook's largest support cluster includes many 'lite' clients (e.g., 'lite' clients for reducing functionality into a simpler, more light-weight or low-low-bandwidth, interface). Such apps typically reinterpret platforms by reducing functionalities.

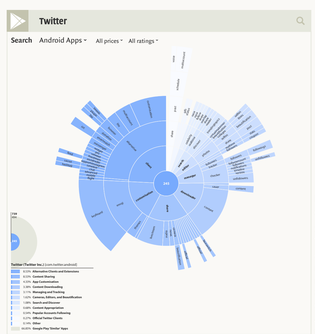

Figure 2b. Sunburst diagram for [Twitter]
Sunburst diagram for Twitter's app support ecology. 245 out of 739 results qualified as relevant, and 8.53% of these are 'Alternative Clients and Extensions'. About 8.53% accounts for 'Content Sharing' apps.

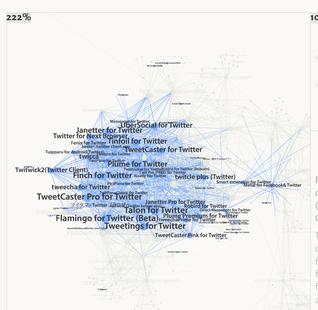

Figure 3b. Network diagram for [Twitter] (zoom: 'Alternative Clients and Extensions')
Twitter's largest support cluster includes apps for enhancing existing functionalities (e.g., for professional and multi-account users, and for strategic use practices).

Conditions for app development

In contrast to Facebook's and Twitter's ecologies, very few alternative clients or content readers have been found for Instagram and Snapchat. This may be explained in part by the increasingly restrictive API regulations of both platforms. For instance, Instagram has discontinued its support for alternative clients for feed reading in November 2015 (Instagram, 2015; Constine, 2015). And in case of Snapchat, the platform does not offer official APIs at all, meaning that access to platform data and functionality are mediated primarily by unofficial or third-party APIs. Similarly, we have found very few content aggregation and search-related apps for these platforms, which would potentially enable users to explore content via alternative modes of sorting and ordering (e.g., chronologically or by topic). In case of Twitter, we are specifically missing the presence of automation and bot-related apps. This is striking, since it is well-known by developers to be one of the ways of relating to Twitter (e.g., via the creation of automated accounts and scripts). This particular developer practice seems to be primarily Web-based, and not app-based.

Discussion

Negotiating platform openness / closure

Social media platforms may incentivise developers to explore and experiment with the 'interpretative flexibility' of their data, user content, and features, but only do so within the limits that are in line with the platform's own objectives or interpretations. API rules and regulations pose a central means for platforms to negotiate their openness / closure towards distinct use cases or user types, raising questions about *how* and *when* platforms seek to either *stabilise* or *open* their platform for innovative interpretations. In addition, further historical research on platform and app support ecologies is required to identify different strategies for how platforms have negotiated their openness / closure towards diverse stakeholders in the past (e.g., other platforms and websites, developers, businesses, and end-users), and why they typically tend towards closure over time.

Towards a stakeholder politics

While previous research has mainly focused on the ways in which platforms negotiate their relations with other stakeholders via API politics (e.g., Bucher, 2013), this study takes seriously the role of developers as central actors in shaping app–platform relations. Such a perspective allows for the exploration of what we refer to as stakeholder politics, or the ways in which developers continuously twist and tweak platform data and functionalities in order to make them fit their own alternative objectives and valuation regimes. In doing so, developers do not only negotiate their own relation to platforms, but also the relations between end-users and platforms, as they often respond to, transform, and rely on their practices of use.

In some cases, developers of apps may also overstep the regulatory limits or conditions that platforms have set. For instance, apps for reposting Instagram content and content downloading apps for Snapchat both overstep such regulations. However, developer ecosystems also remain closely observed by the platform owners for purposes of regulation, but also for internal development and feature integration. In our view, platform politics and stakeholder politics are both inextricably entwined, as both emerge in tight relation to each other.

This empirical investigation interfaces apps with platform studies. First, it contributes to the study of mobile apps by providing a novel empirical methodology for mapping app–platform relations and thereby providing an account of apps as software entities that are both *situated* (existing 'in context') and *distributed* (both shaped by and shaping relations to platforms and diverse stakeholders). Second, it also contributes to the study of platforms by offering insights into stakeholder politics and practices, which we argue are crucial to understanding the defining features of platforms: their *programmability*, *distinct affordances*, *multiplicitous stakeholders*, and *strategies for negotiating* openness / closure.

Alternative approaches

The exploratory study outlined here demonstrates at least one way to study app–platform relations and support ecologies. The key methodological challenge however, is to develop this approach into a more widely applicable framework for empirical app studies. In previous research we have explored two alternative approaches to the study of apps and their ecologies, both of which share part of this method, and departed from the same app store, namely Google Play:

1. First, we have developed a *thematic or issue-based approach*. Following this approach we have mapped networks of apps related to religion queries designed around each of the five major world religions (i.e., Christianity, Buddhism, Islam, Judaism, and Hinduism). We experimented with apps as proxies or indicators of cultural difference and religious specificity. Most significantly, we have found that mobile apps are increasingly used as part of *performing a religion* (e.g., examples in the Islamic sphere include Mecca compasses and prayer time apps), pointing to what we refer to as the *technicity of religion*, or the becoming technical of religion.
2. Second, we have developed a *genre-based approach*. Following this approach we have mapped networks of related apps in order to examine particular genres of apps (not to be confused with the categories provided by app stores themselves), in this case the genre of secure / encrypted messaging apps. Such an approach enables us to study issues of privacy, security, and encryption specifically within the *messaging app sphere* (as opposed to an *issue-demarcated sphere*). Subsequently, we are able to analyse in detail how apps are positioning themselves in relation to current issues and concerns (as with the Snowden revelations in June 2013) or in relation to specific discourses (such as the perceived trade-off for users between usability and security), using textual descriptions written by the developers themselves. Further research along these lines also includes analyses of specific encryption protocols or standards mentioned as features in these descriptions. #

Notes

1. 'Digital Methods for App Analysis: Mapping App Ecologies in the Google Play Store.' *Digital Methods Summer School 2015*. Available from: `https://docs.google.com/presentation/d/1nBwHDtru-0P4fGOVfdT6oG7vCjdQNKcGNUo4tCVvrIQ/edit?usp=sharing`.
2. 'Mapping Secure / Encrypted Messaging App Ecologies,' *Digital Methods Summer School 2016*. Available from: `https://docs.google.com/presentation/d/1fqimqlW64WLSC08TpotTMNjXHqaRqRlNhvuuqgZ0zU/edit?usp=sharing`.

References

Alhar M, MacMahon M, Jha N and Arasu R (2016) App Discovery With Google Play, Part 1: Understanding Topics. *Google Research Blog*. Available from: `https://research.googleblog.com/2016/11/app-discovery-with-google-play-part-1.html` (accessed 19 November 2016).

Bucher T (2013) Objects of Intense Feeling: The Case of the Twitter API. *Computational Culture* (3). Available from: `http://computationalculture.net/article/objects-of-intense-feeling-the-case-of-the-twitter-api`.

Constine J (2015) Instagram Kills Off Feed Reading Apps. *TechCrunch*. Available from: `https://techcrunch.com/2015/11/17/just-instagram/` (accessed 19 November 2016).

Digital Methods Initiative (2016) Google Play Similar Apps. *Digital Methods Initiative Wiki*. Available from: `https://wiki.digitalmethods.net/Dmi/ToolGooglePlaySimilar` (accessed 19 November 2016).

Instagram (2015) Instagram Platform Update. *Instagram for Developers*. Available from: `http://developers.instagram.com/post/133424514006/instagram-platform-update` (accessed 19 November 2016).

Pinch TJ and Bijker WE (1987) The Social Construction of Facts and Artifacts: Or How the Sociology of Science and the Sociology of Technology Might Benefit Each Other. In: Bijker WE, Hughes TP, and Pinch TJ (eds), *The Social Construction of Technological Systems*, Cambridge, MA: The MIT Press, pp. 17–50.

Richter F (2016) Chart: The Most Popular Apps in the World. *Statista*. Available from: `https://www.statista.com/chart/5055/top-10-apps-in-the-world/` (accessed 19 November 2016).

Rogers RA (2013) *Digital Methods*. Cambridge, MA: The MIT Press.

Designed by Fernando N. van der Vlist (`fernandovandervlist.nl`)
Type set in FF Quadraat Sans Display, FF Quadraat Head (titles and headings), and FF Quadraat Sans at 18 pt or 14 pt font size / 21.6 pt or 16.8 pt leading (text)