International Journal of Communication 20(2026), 1780–1798 1932–8036/20260005

App Time: The Dual-Layered Logic of Appification

ESTHER WELTEVREDE
ANTHONY BURTON[1,2,3]
University of Amsterdam, The Netherlands

This article introduces the concept of app time to describe the layered temporalities that structure user interaction and backend operations in mobile applications. Using Tinder as a case study, we develop the networked walkthrough—a novel methodological approach that combines research personas with network traffic analysis—to empirically trace the dual rhythms of user affective experiences and backend processing. We highlight how apps orchestrate user experiences by staging immediacy through continuous backend validations. These coexisting, but divergent rhythms exemplify the dual temporality of app time, in which experiential and computational time coalesce. Our findings reveal how mobile apps stage user experience as performance, creating a sense of seamless interactivity while relying on continuous backend validation. In doing so, this study expands existing theories of appification and software temporality, offering new tools for analyzing how everyday activities are structured, paced, and experienced through digital interfaces.

Keywords: app studies, Tinder, walkthrough method, platform studies, software studies, temporality, digital methods

Esther Weltevrede: e.j.t.weltevrede@uva.nl
Anthony Burton: a.g.burton@uva.nl
Date submitted: 2024-04-02

[1] The authors thank the participants of the SSHRC-funded project *Beyond Verification: Authenticity and the Spread of Mis/Disinformation*, as well as those of the pilot study *The Research Persona as Digital Method*, conducted during the 2019 Digital Methods Summer School at the University of Amsterdam. We also benefited greatly from presenting and discussing this research at the conference *Figurations: Persons in/out of Data* (Goldsmiths, UK, December 2019) and at the ICA preconference *Mis/disinformation and the Artifices of Authenticity and Authentication* (York University, Canada, 2023). We are especially grateful to Wendy Chun for her involvement in the early empirical and conceptual stages of this work, and to Liliana Bounegru, Ganaele Langlois, and Jonathan Gray for their valuable feedback on its conceptual development. Finally, we thank the editors and anonymous reviewers for their generous and constructive comments.

[2] Parts of this work were supported by the Social Sciences and Humanities Research Council of Canada (SSHRC) under a Joseph Armand Bombardier scholarship (Burton).

[3] Authors contributed equally to this article.

Introduction: Orchestrating Experience

Tinder has become emblematic of a new grammar of courtship. Its signature interface gesture—the swipe—has not only defined the app's identity but has also reshaped how users experience, perform, and understand romantic interaction. Imitated by numerous competitors, this seemingly simple mechanism conceals a complex technical architecture that reformats embodied encounters into app-specific actions. Within this architecture, every user gesture is a moment of affective concern. These gestures are only one side of the story. Behind the interface lies a distributed system that structures, validates, and synchronizes these interactions through orchestrated backend operations, including real-time data exchange and profile matching algorithms.[4] This article explores how Tinder's backend processes stage immediacy and coordinate interactions. We argue that Tinder's operations depend on a dual dynamic: on the one hand, affective, embodied interactions unfolding at the interface; on the other, backend responses operating behind the scenes. At the interface, actions such as swiping and tapping occur within an experiential, felt temporality. Simultaneously, these gestures are logged, validated, and processed through continuous backend operations. We conceptualize this mutual constitution as app time—a recursive temporality structured by both user gestures and network rhythms.

To explore this dynamic, we introduce the networked walkthrough, a hybrid method that combines interface-level walkthroughs (Light, Burgess, & Duguay, 2018) with network traffic analysis (Weltevrede & Jansen, 2019). Central to this approach is the use of research personas (Bounegru, Devries, & Weltevrede, 2022)—engineered profiles designed to simulate everyday user behavior. These personas enable us to examine how specific interactions generate distinct patterns of network activity and reveal the operational underpinnings of everyday app use. While our primary focus is temporal, the authentication process plays a key methodological role. Authentication functions as a moment when user actions are recognized and validated across the interface and backend. It is both a user-facing experience (logging in, swiping) and a backend protocol (packet exchange, session refresh). These validations produce the rhythms and delays through which app time is structured.

Following a digital methods approach (Rogers, 2013), we treat authentication not only as an analytical object but also as a research tactic—tracing how backend processes validate, synchronize, and structure user interactions with the front end. In doing so, we build on Agre's (1994) concept of the "grammar of action," which emphasizes how software environments structure user behavior through anticipated sequences of actions. We extend Agre's idea by showing how app architectures do not just anticipate, but rhythmically orchestrate interaction across temporal layers. This allows us to understand app-based platforms not just as cultural interfaces, but as infrastructures of time and coordination.

The next section situates our framework within the broader literature on appification and the temporality of software. We then detail our methodological approach and illustrate it through a case study of Tinder. We conclude by discussing how the orchestration of experience between front-end

[4] Tinder's monetization strategies (subscriptions, Super Likes, and native advertising) are well-documented aspects of its business model (Wilken, Burgess, & Albury, 2019).

https://doi.org/10.65476/bd4srf52

engagement and backend processes reconfigures both the experience and the temporal dimensions of app use.

App Time and the Everyday

Appification refers to the process by which routine activities, social practices, and cultural interactions are translated into mobile applications. It identifies "a historically specific moment when an increasing number of everyday activities and routines are being expressed through, carried out by, and experienced as apps" (Morris & Murray, 2018, p. 3). This process is bidirectional. Activities are first formalized and systematized at the level of the interface, eventually becoming datafied as they are transformed, distributed, and reconfigured within the infrastructures that power apps (Mejias & Couldry, 2019). Appification simultaneously reformats embodied actions at the interface and integrates them into expansive backend operations. As Distelmeyer (2019) observes, contemporary computing marks a shift from object-oriented interaction, centered on discrete files, to process-oriented engagement, in which interfaces operate as conduits for continuous technical and ideological processes. Seen through this lens, meaning emerges and circulates in terms of a data-infrastructural perspective, not as the assignation of value to a particular event, but through a network of points of contact that distribute "agencies and relationships between different categories of communicational actors," where they "participate in the organization of a broader network of power formations" (Langlois, 2014, p. 69). While much of the existing literature has focused on front-end processes—how everyday interactions are translated into embodied, formalized gestures—our contribution emphasizes how these activities are processed, transformed, and distributed through backend app architectures. Building on these ideas, we further explore how appification operates on two intertwined levels: one rooted in the immediate, embodied interactions of users and the other in the underlying backend systems that support these apps.

Figure 1 visualizes appification as a dual-layered process unfolding across the front end, the app interface, and its supporting backend systems. On the top, the user initiates action through affective, embodied gestures–swipes, taps, and other micro-interactions—which are shaped by everyday habits and engagement rhythms. These gestures are passed to the app's user interface, which functions as a mediating layer: It translates user actions into structured outputs, offers functions and preferences for interaction, returns algorithmic outputs, and synchronizes user states across time. This coordination of felt experience with computational logic enables the phenomenon we theorize as app time. From the app interface, data flows to the backend architecture and network exchanges, where core operations, such as data storage, state retrieval, and functionality execution, occur. Authentication and validation—the process by which gestures are recognized, logged, and responded to—occur between the app interface and the backend. This transitional operation ensures that each interaction is acknowledged, enabling recursive progress through the interface. Third-party systems support functions such as analytics, ad delivery, and external data synchronization. These systems further extend the scope of appification by contributing real-time monitoring and computational feedback that modulate both backend and user-facing experiences. All communication between the app, the server, and third-party systems takes the form of network activity—a key empirical trace that we captured through the networked walkthrough method. Together, these layers and connections constitute a recursive and

asynchronous system for staging digital interactions, capturing the layered temporality of appified environments.

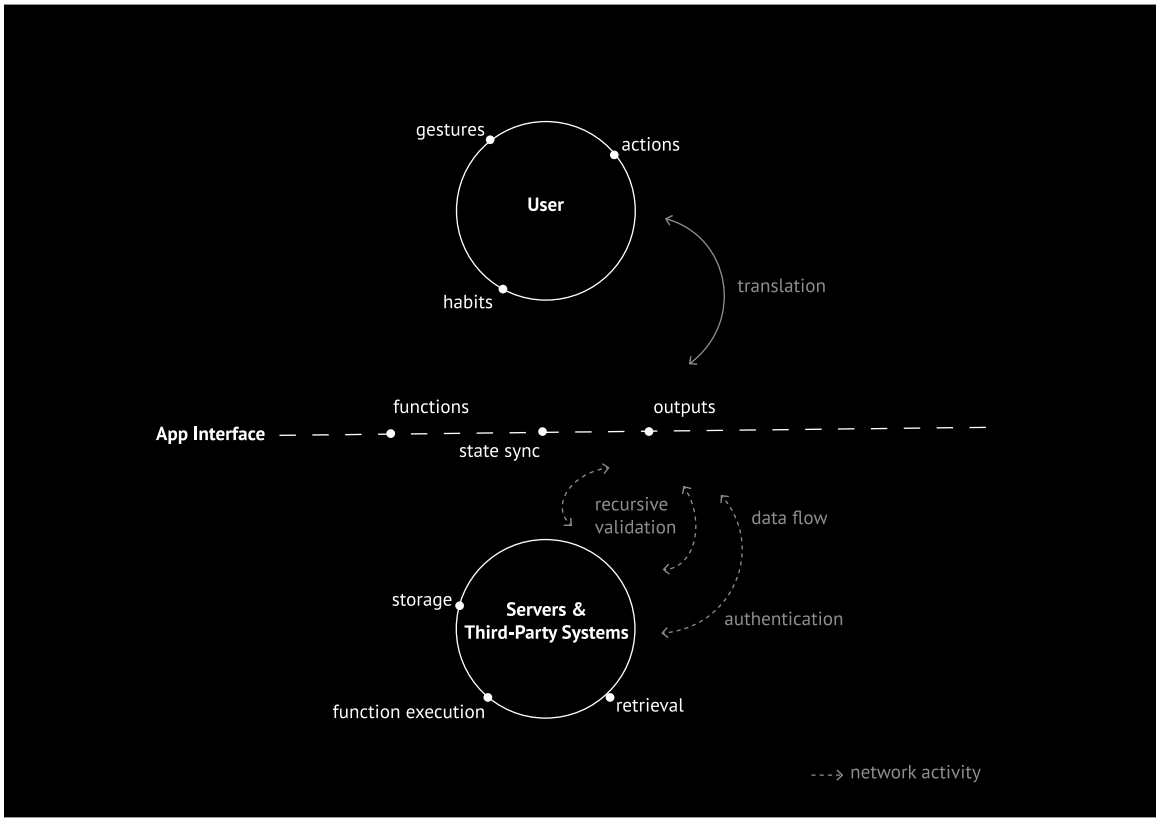

Figure 1. Dual-layered appification. The figure schematically represents the reciprocal relationship between front-end gestures, the app interface, and backend processes, highlighting how interaction and validation occur across these interconnected layers. Visualization by authors.

Central to these dynamic interactions is what Agre (1994) terms the "grammar of action." Grammars of action are how representational systems "employ formal 'languages' for representing human activities," thereby treating these activities "as a kind of language itself" (Agre, 1994, p. 108). Grammars of action underlie Agre's (1994) capture model of surveillance (p. 101). They structure human activities and subsequently reorganize them for schematic representation. Users engage with apps through actions like pressing, swiping, or other context-specific gestures organized into systematic grammars that guide and authenticate user behavior. Appification thus describes how everyday activities are reformatted through app-specific vocabularies. Dieter and Tkacz (2020) extend this idea by describing "appification" as the process through which everyday activities are instrumentalized through software, "where new equivalences can be observed as well as new points of differentiation" (para. 2). In their study of "challenger" banking apps in the UK, they describe how banking apps subsume the formerly "everyday" activities that comprise interaction with a bank into new grammars that, in turn, structure and verify user

https://doi.org/10.65476/bd4srf52

identity and behavior. Behavioral patterns emerge from these new grammars as actions become instrumentalized through repeated encounter between the user and the app. This interaction is integral to how Tinder configures user experience, allowing it to communicate authenticity by translating embodied actions into digital credentials (Duguay, 2017, 2019), despite the "presumed" nature of this mode of intimacy (David & Cambre, 2016).

Appification is inherently bidirectional. While front-end appification focuses on the embodied, immediate practices of users, backend appification encompasses the transformation of these practices into data and their orchestration through the app's operational systems. This includes the complex networks of advertising platforms, content delivery networks (CDNs), advertising networks, data analytics, and software development kits (SDKs) that collectively support modern apps (Chao, van Geenen, Gerlitz, & van der Vlist, 2024; Lai & Flensburg, 2021; Lomborg, Sick, Flensburg, & Lai, 2024; Pybus & Coté, 2024). Within this operational frame, user actions are continuously recognized, preempted, processed, and modulated through backend systems, creating a recursive rhythm of interaction that extends beyond the interface. In this way, everyday gestures and routines are subsumed into a larger data-driven ecosystem, further structuring the tempo and logic of the appification process. This dual-layered orchestration echoes the insights found in Sal Hagen's dissertation. Hagen's analysis of "reactionary rhythm" on 4chan's/pol/ (Hagen, 2024) demonstrates how repetitive digital practices are temporally structured, revealing a complex interplay between immediate user actions and the underlying digital flows. Similarly, our analysis of Tinder shows that while a swipe may immediately contribute to a user's flow state, the corresponding backend processes occur with distinct timing and rhythm.

On the front end, Tinder appifies romantic encounters through tactile actions such as swiping, which translates an act of visual judgment into a patterned, embodied interaction. Each swipe fits into a preestablished grammar of action, coordinating user action with the app's timing and engagement structures. The swipe is just one of multiple aspects, such as profile creation, that draw users into particular subject formations (Berger, 2023; Duguay, 2019; MacLeod & McArthur, 2019. Tinder's backend processes these interactions by integrating them with CDNs, advertising networks, and data analytics systems. These actions are then absorbed into a sophisticated data-processing infrastructure (Weltevrede & Jansen, 2019) where they are aggregated and analyzed, forming the backbone of an environment of expected use (Light et al., 2018). This convergence of embodied actions and computational processing not only coordinates user engagement but also shapes the experiential dimension of app use in what Striphas (2015) identifies as the algorithmic conditioning of habitual engagement. In particular, the repetitive, tactile gestures involved in front-end appification can give rise to what Mihaly Csikszentmihalyi (2013) terms a "flow state," defined as the affective state of being "in the 'zone'"—fully immersed in the task at hand, without the influence of the outside. Studied initially and attributed to the concept of human creativity and play—Csikszentmihalyi's interview subjects, when conceptualizing flow, were "artists, dancers, surgeons, and basketball players" (Soderman, 2021, p. 3)—the process of flow has been adopted by software designers as a goal state for users, wherein attention is focused entirely and the subject is absorbed in the play of the task at hand. What begins as a series of routine swipes on Tinder, for example, can evolve into a state where users become

entirely engaged with the app, effectively producing a rhythm of attention that merges affective presence with computational pacing.[5]

This affective engagement and recalibrated temporal experience is a central outcome of appification: Repetitive gestures and the timing of their orchestration produce new temporal regimes. Research on the temporal regimes created through apps configures user time by prioritizing relevance, personalization, and engagement over linear, clock-based time. By engaging users through affective responses, apps shift user operations from real-time logic to what Taina Bucher (2020) calls "right-time" logic. In this model, content is delivered in a manner that reflects shifting patterns in user behavior, creating a sense of ideal timing, or what Bucher describes as the "kairologic" (from the Greek *kairos*, meaning "opportune time") of algorithmic media. For example, Facebook's news feed uses factors like user engagement patterns, device usage, and time of day to determine when to show advertisements or resurface old posts. When algorithms get the timing "right," Bucher argues, users may experience serendipity or connection—but they can also feel discomfort and even alienation as they navigate the conjunction between clock time and algorithmically determined "right time" (Bucher, 2020, p. 1709). This tension is further developed by users' active modulation of their digital temporality. Through interactions such as "conscious clicking," users attempt to realign their personal sense of timing with app predictions (Bucher, 2020, p. 1711), even as algorithmic logic—shaped by the app's goal, user behavior, and technical infrastructure—seeks to maximize engagement and monetization. Jacobsen (2024) extends this discussion with his notion of "anniversary logic," whereby apps use personalization and data profiling to resurface memories at emotionally resonant moments, embedding themselves in users' daily lives through the "rhythms of remembering" (p. 2884). The logic of appification is applied to the human act of remembering and reflecting. By embedding themselves in these affective processes, apps begin to dictate how and when users engage with the past and reconfigure agency through this shift in temporal control. Similarly, Lohmeier, Kaun, and Pentzold (2020) argue that datafication produces a plurality of temporalities, as predictive analytics, for example, anticipate user needs and deliver content at optimal moments, dictating how users experience time and prioritizing corporate interest over individual well-being and health.

The temporality of app use cannot be reduced to a simple matter of user perception or interface responsiveness. It unfolds through a stratified arrangement of rhythms, delays, and anticipatory operations that exceed a singular logic of temporal measurement. Rather, app time is constituted through a dynamic negotiation between embodied interaction (and the different ways that perception manages temporality) and backend calculations. Mark Hansen (2015), through the concept of "feed-forward time," claims that digital media do not merely record or reflect experience; they fundamentally preempt it. In

[5] This state of flow has been problematized in the construction or extraction of user activity and attention. Soderman, for example, argues that flow has both a political and ideological tint, positioning Csikszentmihalyi's lifework on the idea as a possible response to the latter's upbringing under post-WWII Marxism. Soderman's critique draws from anthropologist Natasha Dow Schüll's study of the design of Las Vegas casinos to encourage addictive behaviors. As Schüll (2012) writes, flow has a "desubjectifying" effect, wherein the user is "carried forward" by the infrastructure that they have become absorbed in (p. 166, quoted in Soderman, 2021).

https://doi.org/10.65476/bd4srf52

his view, contemporary computational systems operate under a feed-forward logic, in which vast sensor-based and data-driven infrastructures anticipate future actions before they enter conscious awareness (Hansen, 2015). This reorients temporality away from phenomenology and toward operativity. Time becomes not what is lived or recalled, but what is acted upon by algorithmic systems in advance of subjective experience. In Tinder, this logic manifests in how data are prefetched, user traits are modeled, and advertising content is selected long before users actively engage with its presentation. Instead, the user experiences apparent spontaneity that is computationally staged yet remains undetectable because it operates on a different temporal register. Since network operations occur on a scale of microseconds, beyond the reach of human perception, this register exists outside of the traditional humanist modes of constructing and making sense of time—namely, narrative and memory (Ernst, 2016). The operations of code, circuits, and signal timings produce a temporality that requires alternative ways of accounting for time beyond synchronicity or delay. Rather than being out of sync with human time—any discussion of synchronicity implies the possibility of shared measurement—these temporalities are machinic and recursive; in short, indifferent to human rhythms. When apps like Tinder load and unload profile data or poll servers for matches, they do so on a temporal scale that is at necessary odds with any discrete understanding of this time, let alone the state of flow manufactured through the gesturalization of the everyday.

Appification, then, becomes a grammar for everyday life by substituting its own procedural flows for quotidian processes that are otherwise understood as discrete moments in time. Manovich (2013), in *Software Takes Command*, argues that software itself is always a performance at the interface level. By preempting discrete temporal processes, appification replaces them with its own software-based performance. As Manovich writes, software is not simply a tool or object, but a process that unfolds through interaction; this leads to the affective and sensory appeal of appification, where the interface invites a user into a state of fluidity that is itself tightly managed through the discrete chronology engineered by the backend. App time is thus a doubling and recursive temporality that exceeds both human-centered models of immediacy and traditional accounts of asynchronous media. It is a mutual staging of perception and computation, actively shaped by anticipation and affective experience. It is both the temporal logic of the machine and the temporal feeling of the user, woven together in ways that reflect broader shifts in how digital life is paced, sensed, and lived.

In the case of Tinder, we observe that the repetitive swiping actions inherent in its design do not merely confirm user identity, but structure experience itself. Tinder's design—where users swipe left or right on a potential match's image—can quickly lead to a particular flow state in which rapid, embodied gestures not only produce continuous interaction but also reshape the user experience of time. This flow state is a patterned effect of how temporal logic, interface grammar, and computation converge, resubjectifying the user in accordance with the application's temporal logic. As these processes reshape daily practices, they call for new ways of accounting for how gestures, rhythms, and computational operations coproduce everyday digital life. In the subsequent sections, we present a variation on the walkthrough method (Light et al., 2018)—integrating insights from network traffic research—and present an empirical case study to demonstrate this extended perspective on appification, drawing on concepts of grammars of action and app architectures to illuminate the temporal dimensions of app use.

The Networked Walkthrough Method

To analyze the layered temporalities of app time, we introduce the networked walkthrough: a methodological approach that integrates user-centered interaction analysis with backend network observation. This builds on approaches to "multi-situated app studies" by identifying key "entry points" for analyzing apps (Dieter et al., 2019). At the interface level, our method draws on the development of research personas to study personalization and positionality in digital research. Research personas serve as analytical devices designed to understand the multifaceted means by which different layers of digital networked media perceive online users. It allows for the investigation of "flows of information that work to configure the experience of individual users who interact with various content online" (Bounegru et al., 2022, p. 91). Unlike macro-level API-based analyses that adopt a "view from above," the research persona offers a user-centered perspective, highlighting the reciprocal relationship between algorithmic flows and the user's subject position.

In our networked walkthrough, we use research personas as a methodological device. Following Bounegru et al. (2022), we employ personas to deliberately explore algorithmic personalization and the cultural logics embedded within apps. These personas are constructed with fictional backgrounds and interaction patterns designed explicitly to provoke responses from the Tinder algorithm and its data-collection mechanisms. In this way, personas function methodologically by systematically structuring and focusing our empirical inquiries on specific experiential and algorithmic dynamics. The research persona method complements the walkthrough method in app studies, a systematic process of "walking through" the user flow of an app to document the app's features. Initially developed for software engineering and user experience design, it was adapted by Light et al. (2018) to study applications through the lens of humanistic and scholarly inquiry. Weltevrede and Jansen (2019) advance the possibility that the walkthrough method can be used to identify data flows and interactions between users and apps that are established through the usage relationship. Research personas extend the walkthrough method by enabling researchers to examine how the usage flows of apps are expressed differently for different user categories and how personalization processes influence user interfaces. In our study, the research persona surfaces how users are authenticated as users through the flow of the walkthrough itself—we gained experiential knowledge from setting up the research accounts and observing personalized content, such as advertisements, geared toward the account's preferences.

The networked walkthrough functions as a methodological approach. As a method, it involves the systematic exploration of an app's interface and the underlying data infrastructure, tracing interactions from front-end actions (e.g., swiping or messaging) to backend processes (e.g., data collection and algorithmic matching). The networked walkthrough operationalizes the dual-layered model of appification by making visible the particular relationships between user-facing gestures and backend processes. This alignment between method and framework builds on Duguay and Gold-Apel's (2023) argument that the walkthrough can function as both an investigative practice and an interpretive lens, surfacing the cultural and technical logics embedded in app behaviors. At the network connection level, our approach extends the work of Dieter et al. (2019), who discuss how app packages and dynamic network connections reveal the data flows that draw users into a networked ecosystem.

https://doi.org/10.65476/bd4srf52

While the walkthrough method has historically engaged with the material and contextual aspects of applications and their user experiences, it remains unclear whether the method can effectively "bracket" users from the larger platform ecosystem that applications cultivate between the researcher and other users—for example, the ethical considerations and potential harms to other users arising from the persona's participation in media spaces. While this ethical question arises through our research design (in the context of real user profiles encountered during our walkthrough of Tinder), our matching of two user personas together to study network activity partially elides this. It particularly aims to examine the application's deep network activities while minimizing opportunities wherein users must be bracketed away from encounters during the researcher's walkthrough.

To compare user experience with network rhythms, we designed a walkthrough method that focused on two core events in Tinder: the swipe and the match. We created two research personas—Charles (male, iOS) and Nora (female, Android)—and engineered a scenario in which they would match one another. Using network monitoring tools, we captured all network-layer actions corresponding to their interactions, including when one swiped right on the other and when the match was mutually confirmed. Using age and location parameters, Nora's search settings were engineered to exclusively encounter Charles's, while Charles's persona swiped left on other profiles before ultimately swiping right on Nora and triggering the match event.

For the network analysis, we used two tools: AppTraffic (Chao et al., 2024) and the Charles debugging proxy application. Both tools function similarly by setting up proxies that allow interception and examination of secure HTTPS from the iOS phone and encrypted network traffic from the Android phone.[6] This setup enabled us to capture and analyze the data transferred from the app to various server endpoints. The focus of our network analysis was twofold. First, we produced an event-aligned map of network activity, tracking when requests occurred, where they were sent, and how these transmissions corresponded to interface activities (e.g., swipes, matches, ad views). Second, we characterized the traffic itself, examining request types, endpoints, and temporal patterns (e.g., authentication pulses, prefetching, analytics bursts) to relate front-end engagement to backend processes. Taken together, the mapping and characterization provide a comprehensive view of app authentication that spans user-facing events and machine-oriented operations.

The outcome of our walkthrough is the visualization shown in Figure 2. This diagram captures the dual-layered temporal orchestration between front-end actions and backend system-level responses. Along the horizontal axis, the timeline records micro-interactions from user profiles being viewed and swiped to matches and advertisements. The top half represents interface-level events—the user's experience of flow—while the bottom half represents backend processes triggered by those gestures. Some backend events are immediate, while others (ad loading, fast match requests, and third-party analytics) lag or occur outside of the visible user session.

[6] This discrepancy is due to changes in how Android handles trusted certificate authorities since 2016 (Google).

The visualization uses a symbol-based legend to clarify the meaning of each element. Purple circles represent POST requests, while orange circles indicate GET requests. Media assets are shown using distinct icons: filled squares for profile images, open squares for Spotify covers, and diamonds for Instagram images. Dashed arrows trace when a profile element is loaded in the backend and only later surfaced on the front end. Names label profiles that were loaded, but displayed later, while names in orange text identify those that were loaded, but never shown to the user. This visual encoding helps demonstrate the layered, asynchronous operations that underlie seemingly seamless app experiences. In this way, the networked walkthrough reveals temporal disjunctions, showing when front-end events are confirmed or deferred by backend systems. It also reveals which backend activities remain entirely hidden from users, even as they shape the rhythm and pacing of the app experience.

The networked walkthrough reframes the study of apps not only as interface phenomena but also as networked systems. In doing so, the method allows researchers to trace the temporal disjunctions between the user's flow state and the network rhythms of app logic—what we call app time. This dual-layered method does not seek to fully resolve the opacity of platform infrastructure, but rather to situate users' lived experiences within the machinic patterns that shape and respond to them.

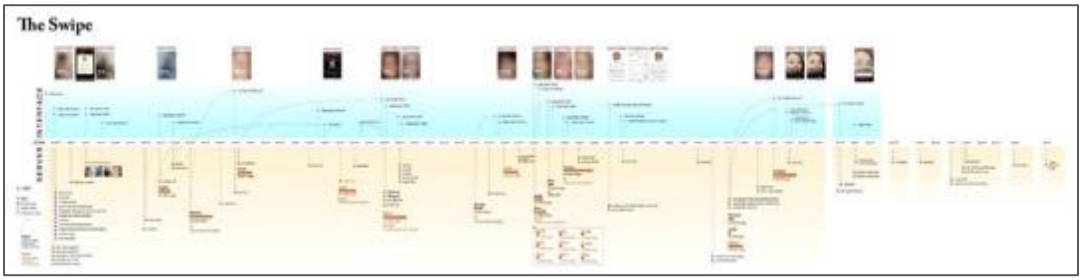

Figure 2. The swipe–timeline of front-end and backend activity while using Tinder. The figure aligns user gestures (swipes, matches, ads) with backend processes (HTTP requests, media loading, analytics), revealing the layered orchestration of app time. Visualization by Carlo de Gaetano.

Temporal Orchestration and Layered App Time

Our analysis follows Light et al.'s (2018) walkthrough method, which delineates three distinct phases of study: "registration and entry," "everyday use," and "app suspension, closure and leaving" (pp. 892–894). First, we examine front-end engagement during key phases of app use—specifically "registration and entry" and "everyday use." Second, we explore backend processes by studying the network-level communications that underpin the app's operation.

Front-End Temporality and Affective Gesture

Registration and Entry

We begin with the registration and entry phase, during which users set up accounts and make their first contact with the app. Each persona was assigned a prepaid phone number and separate device—Charles

https://doi.org/10.65476/bd4srf52

on an iPhone (with preexisting app history) and Nora on an Android phone (with a newly created Google account). Their Tinder profiles were created using AI-generated profile photos and customized metadata: Nora was set to 72 years old with a narrow search radius (1 km), while Charles was set to 32 years old with a preference for older users. This setup minimized interference from real profiles and enabled us to control when and how the match occurred. While registration traffic was monitored, the majority of data capture occurred during the "everyday use" phase.

Timing is a central focus of this analysis. The flow state that Tinder engenders—through the repetitive act of swiping—builds momentum and shapes user engagement. By capturing network packets, we sought to identify when and how the construction of this flow state might diverge from backend temporalities.

Everyday Use

Next, we turn to front-end gesture-based engagement, focusing on the "grammars of action" that mediate interactions with Tinder. These grammars—embodied in gestures like swiping—both guide user behavior and serve as mechanisms for authenticating user actions. We analyze the action of swiping, a fundamental interaction at the heart of Tinder's user experience. To provide a comprehensive understanding of this process, we developed a detailed timeline (Figure 2) that synchronizes front-end events (e.g., visible interactions) with backend processes (e.g., data loads from cloud servers). The upper half of the timeline shows when elements become visible to the user, while the lower half indicates when these elements are transmitted or loaded from the backend, supplemented with screenshots capturing moments of active engagement.

Swipes and interface interactions generate bursts of backend processes, but these responses are not always immediate or symmetrical. For instance, advertisements or matches are often delivered with a delay or interspersed with third-party responses, such as fast match teasers and analytics events. This suggests that backend processes continue even after the user has moved on. Second, we observe "temporal drift," where profiles or assets are loaded on the backend before or after they appear on the interface. This decoupling of front-end flow and backend logic illustrates the asynchronous nature of app time. Third, backend activity is not evenly distributed: Seemingly insignificant gestures like a "nope" swipe on an advertisement or profile can generate significant data-rich moments (e.g., loading ads or batch profiles), whereas events like receiving a match result in relatively sparse data transmission. These discrepancies show that data intensity is not proportional to the affective intensity of user experience. When a user swipes on Tinder, several backend processes are triggered almost instantaneously. However, these processes are not always immediately reflected on the front end. This delay, whether intentional or due to processing time, plays a crucial role not only in how users perceive and interact with the app but also in how the application receives and polls for user data.

The grammars of action are not merely functional; they are carefully designed to shape user behavior and engagement. On the front end, swiping pulls the user into the application's flow; on the backend, it triggers various actions and calls for data from the application's servers. Thus, one of our key findings is that front-end and backend operations unfold on distinct, but interdependent temporal layers

(Figure 3). Profiles are often preloaded by the backend before they are shown to the user. Some profiles, such as Ingrid's, are never displayed, while others appear only after a delay. Dashed arrows in Figure 3 visualize this lag, revealing the asynchronous orchestration of app time. Rich media assets, such as Instagram or Spotify content, are prefetched regardless of whether the associated profiles are surfaced, pointing to a speculative computational logic that loads and prepares content based on predictions that do not always materialize. As Figure 2 illustrates, at 14:45:53, three profiles are loaded from the backend simultaneously, while over the course of the next 14 seconds—and two swipes by the research persona—these profiles appear to the user. The match with the persona Nora likewise illustrates a dual-layered process with the backend: While Nora's profile is swiped right at 14:45:57, the match is not revealed until 11 seconds later, at 14:46:08. This could be attributed to processing on the backend, namely a check with the server to see if the match is mutual, but the speed at which the persona swipes in the flow state means that another profile is loaded before the match with Nora interrupts the flow.

This dual-layered temporality extends to the app's advertising strategies (Figure 4). Ad interactions generate a disproportionately high volume of backend activity relative to their affective weight on the front end. Each swipe on an advertisement triggers multiple backend operations: The system loads banners, requests analytics, and tracks interaction events, even when users swipe past ads almost immediately. Calls to Google DoubleClick and other ad servers are made continuously, with user data shared upon ad interaction and new ads prepared immediately thereafter. Once an ad is dismissed, the application promptly loads the next one. These patterns underscore how monetization and targeting take system-level priority, resulting in more data-intensive processing than more meaningful user events like matches.

We observed that backend data intensity does not align with moments of heightened user engagement. Advertisements, for instance, represent data-rich instances, whereas a match—a moment of heightened emotional salience for the user—generates minimal backend processing (Figure 5). This asymmetry shows that system-level intensity follows infrastructural priorities, such as engagement tracking and ad delivery, rather than affective weight. Seemingly trivial gestures—such as a simple "nope" to reject a profile—could nonetheless be data-rich moments, triggering the loading of multiple profiles and sometimes advertising information. At 14:45:26, for example, a "nope" on an advertisement initiated a new ad request four seconds later and pulled detailed persona information, including preference data, and shared it with the advertiser. Such moments of mundane flow-state activity mask the punctuation of intensive data transfer on the backend.

These observations underscore a fundamental disjunction between the affective flow experienced by users and the app's backend data processing. Collectively, Figures 3 to 5 illustrate three key dynamics: (1) temporal drift, where backend processes preload or delay content display; (2) backend asymmetry, where data-rich operations (e.g., ads) do not align with user significance; and (3) recursive orchestration, in which backend systems shape, pace, and often preempt user engagement based on probabilistic logic. Together, these figures exemplify the core dynamics of app time as a layered temporal performance.

https://doi.org/10.65476/bd4srf52

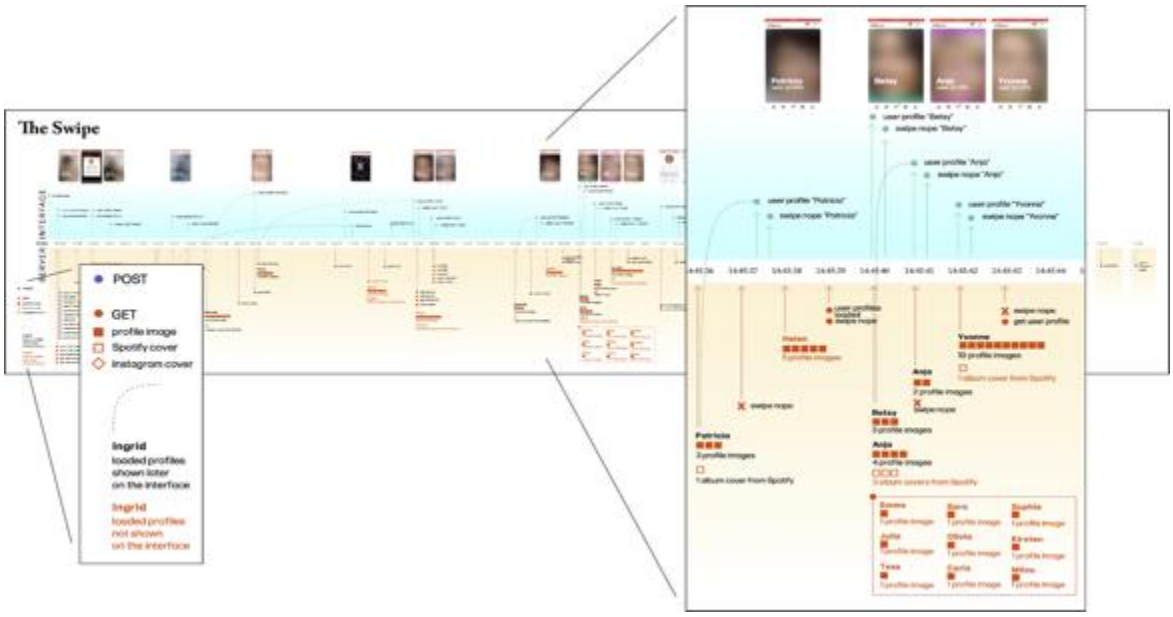

Figure 3. Dual-layered temporality of profile data. The figure shows how profiles are often preloaded by the backend before appearing in the interface—or not at all—revealing the asynchronous orchestration of app time and the speculative logic of prefetching. Visualization by Carlo de Gaetano.

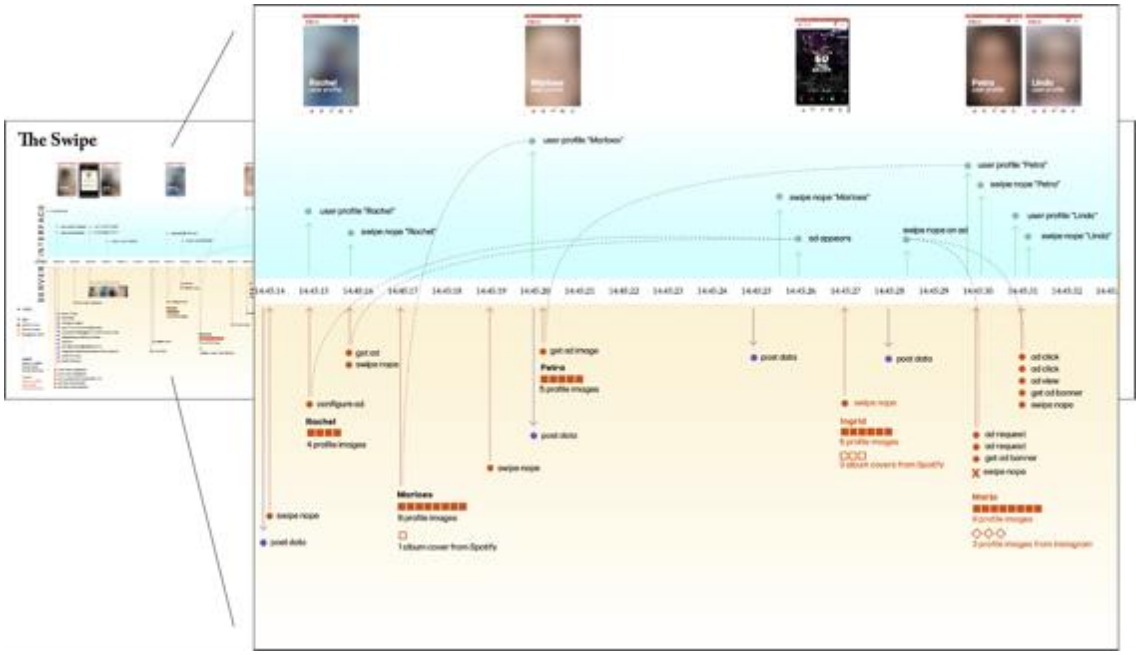

Figure 4. The advertisement as a data-rich event. The figure shows how ad interactions generate disproportionately high backend activity, underscoring platform-level priorities of

monetization and targeting over affective user events like matches. Visualization by Carlo de Gaetano.

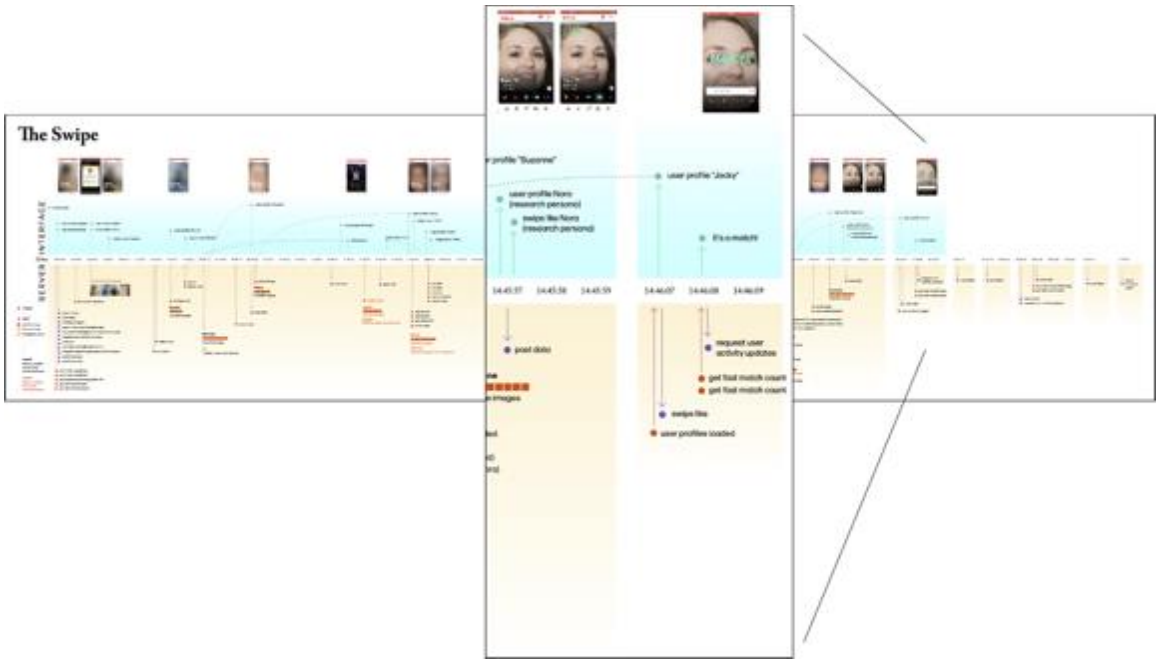

Figure 5. The match as a data-poor moment. The figure contrasts the emotional significance of a match with its relatively minimal backend processing. Visualization by Carlo de Gaetano.

Network Rhythms and Continuous Validation

The second part of our analysis shifts focus to network rhythms—the continuous, network-level processes that support and validate user engagement. Using the Nora persona, we captured network communication flows between the user's device and the servers. Our analysis focuses on the Transmission Control Protocol/Internet Protocol (TCP/IP) model, which categorizes network communications into distinct layers: application, transport, network, and link. Our previous discussion of the HTTP layer is just one facet of this model. Next, we examine rhythms across these layers. Our visualizations present both the state of flow on the front end and backend (Figures 2 to 5) and the interactions across the TCP/IP network layers (Figure 6). Figure 6 makes these network rhythms visible, showing a spike of activity at app launch, a muted response to the emotionally significant match event, and regular backend pulses that continue even without user input. This visualization reveals the continuous authentication and acknowledgment processes necessary for user actions to be enabled and shared across the network. Notably, the opening of the app triggers a high volume of diverse connections across all layers of the TCP/IP stack. This burst of activity occurs before any visible user interaction occurs, signaling the preemptive nature of backend orchestration.

Unlike the connections initiated at the opening of the application, a "match" results in only a few additional connections on the HTTP layer. Likewise, the transport and application layers—those layers that

https://doi.org/10.65476/bd4srf52

deal with transmitting application-level data between the front end and the backend—are more active at the opening of the application, where data is loaded upon login, than during the match. The match event, which is affectively meaningful for the user, appears as a relatively low-intensity backend moment, concentrated in the application layer and shaded red in the visualization. The activity of users does less for the process than the simple activity of opening and keeping the application open. This can be seen in the communications at the network layer level. The network layer, which establishes and deals with packet transfer between devices, is constantly engaged during the use of the application and transmits between devices at a steady rate. Even in the absence of user input, backend traffic continues at predictable intervals—particularly in the network and data link layers—pointing to the autonomous, self-sustaining cadence of system-level verification, status updates and connectivity checks, and app-state maintenance. What is required to sustain appified interaction happens beneath the level of recognizably "user-facing" data. Appification is enacted not only through expressive front-end gestures but also through invisible, low-level maintenance and packet flows that form the substrate of perceived continuity. Instead of being merely reactive, this network connection is proactive and persistent: App and server remain in constant dialogue—"dialing home" repeatedly—to sustain the layered, asynchronous experience of app time. This finding highlights the nuanced relationship between user interactions and the underlying backend processes. The visualization in Figure 6 makes these patterns visible, showing how backend rhythms continue before, beyond, and beneath visible app events.

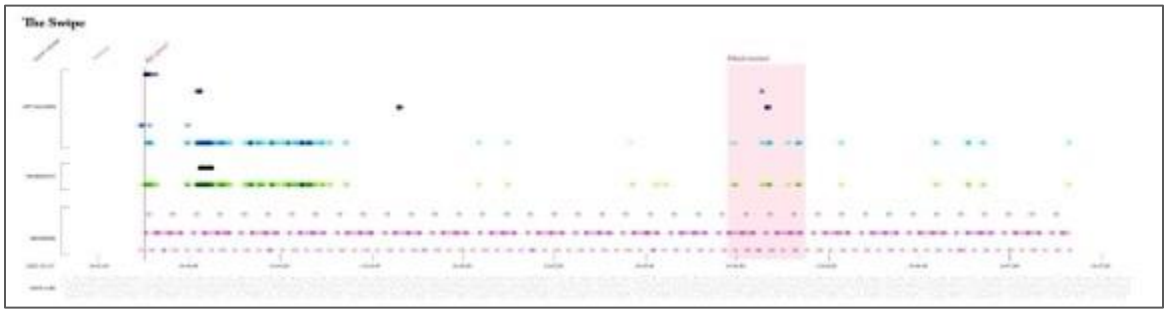

Figure 6. Systems-level continuity across TCP/IP layers. The figure shows system communication across four network layers during Tinder use, with a spike of activity at app launch (red line) and regular backend pulses that persist even without user input. The shaded block marks the moment a match is received—affectively significant, but low-load for the system. Visualization by Carlo de Gaetano.

The Dual-Layered Temporality of Appification

A decade after Tinder revolutionized dating, the app has seamlessly woven itself into the fabric of everyday life, with its core mechanics—swiping, matching, chatting—becoming ingrained in contemporary dating practices (David & Cambre, 2016). Courtship proves fertile ground for how time, experience, and identity can be orchestrated through the differing scales of interactivity when approaching apps. Appification, in this sense, should be understood not only as the transformation of everyday activities into app-based interactions but also as a dual process that operates both on the level of user experience (through embodied, tactile gestures) and on the level of backend architecture (through data flows and network-level

coordination). This duality forms the basis of what we call "app time," a recursive temporal performance in which affective immediacy is staged atop computational timing.

A key contribution of our research is the development of the networked walkthrough method. By integrating research personas with network traffic analysis, this method enables us to capture and compare the temporal and affective dimensions of user interactions with the underlying, often invisible, backend processes. We characterize app time as a dual-layered temporality, marked by distinct yet interconnected rhythms—front-end affective immediacy and backend rhythms, continually co-constituting each other. Our study demonstrates that this temporal disjunction between the immediate, affective experience of the user and the backend's asynchronous response mechanisms is not merely a technical artifact, but a fundamental aspect of how platforms like Tinder reconstruct the everyday. While a swipe may instantly immerse a user in a flow state at the interface, the corresponding backend processes—data loading, advertisement calls, and match verification—occur on a different temporal rhythm. This dual-layered temporality underscores the temporal decoupling of experience and computation, wherein user-centric experiences and system validations coexist yet follow distinct temporal logics. Such gaps—preloading, deferred matches, and real-time polling—demonstrate that app time is not passive lag, but active orchestration, performed by network rhythms beneath user perception. Swiping facilitates a flow state, where the embodied act of interaction becomes rhythmically repetitive and immersive. However, as we have shown, this experience is not synchronized with the backend processes that validate the authenticity of user actions.

App time names the recursion that occurs when the time of desire is preconditioned by the timing of machines that register this desire as data. Such conditions are not neutral. As our walkthrough revealed, the most data-rich moments are not the same as those of emotional payoff. Meaning is calculated differently, preempting how it can emerge. This underscores a broader critique: App time, through its affective immersion, is also a medium of control. The backend does not merely respond to user actions, but preempts them according to logics of engagement and monetization, as Hansen (2015) describes, in a feed-forward manner. The experience of flow is thus as much a product of technical architecture as it is of personal attention. While we sought to articulate app time, our study is limited in its empirical scope. The primary limitation is the restriction to Tinder, which, while exemplary for app temporality, might not generalize to apps with distinct temporal structures or interaction models. Further studies should compare temporal dynamics across diverse apps (banking, healthcare) to extend understanding of appification's broader implications. Additionally, our analysis is based on two research personas interacting within a single app under controlled conditions. While using personas allowed us to precisely track the temporal interplay between front-end experience and backend activity, this approach cannot fully account for the diversity of real user behaviors, preferences, and contexts. Tinder users vary widely in how they engage with the app depending on factors such as age, location, identity, and intention (e.g., dating vs. entertainment vs. networking), and our study does not capture this. On the backend, our network traffic analysis cannot provide full visibility into encrypted processes, third-party tracking behaviors, or the logic of machine learning systems that personalize content behind the scenes. In this sense, our walkthrough is limited not only by its sample but also by the technical opacity of the app ecosystem itself. Future work could expand this study by including a wider range of user types, conducting in-the-wild ethnographies, and integrating reverse-engineering techniques to further demystify the backend.

https://doi.org/10.65476/bd4srf52

In line with our peers in this special issue, our study raises crucial questions about the role that affect plays in understanding authenticity. What feels authentic in app-mediated environments is reconstituted through algorithmically conditioned experiences, affecting how users perceive genuine interactions. The flow state of a swipe may feel spontaneous, emergent from user desire, but it is embedded in a tightly scripted backend architecture that commodifies and datafies this phenomenon. These actions are not merely aesthetic, but computational: The sense of immediacy, the pace of engagement, and the rhythm of reward coproduce this new mode of appified courtship.

References

Agre, P. E. (1994). Surveillance and capture: Two models of privacy. *The Information Society, 10*(2), 101–127. https://doi.org/10.1080/01972243.1994.9960162

Berger, V. (2023). Mediatized love: A materialist phenomenology of Tinder. *Social Media + Society, 9*(4), 1–15. https://doi.org/10.1177/20563051231216922

Bounegru, L., Devries, M., & Weltevrede, E. (2022). The research persona method: Figuring and reconfiguring personalised information flows. In C. Lury, W. Viney, & S. Wark (Eds.), *Figure: Concept and method* (pp. 77–104). Singapore, Singapore: Springer Nature.

Bucher, T. (2020). The right-time web: Theorizing the kairologic of algorithmic media. *New Media & Society, 22*(9), 1699–1714. https://doi.org/10.1177/1461444820913560

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

Csikszentmihalyi, M. (2013). *Flow: The psychology of happiness*. New York, NY: Random House.

David, G., & Cambre, C. (2016). Screened intimacies: Tinder and the swipe logic. *Social Media + Society*, *2*(2). https://doi.org/10.1177/2056305116641976

Dieter, M., Gerlitz, C., Helmond, A., Tkacz, N., van der Vlist, F. N., & Weltevrede, E. (2019). Multi-situated app studies: Methods and propositions. *Social Media + Society, 5*(2), 1–15. https://doi.org/10.1177/2056305119846486

Dieter, M., & Tkacz, N. (2020). The patterning of finance/security: A designerly walkthrough of challenger banking apps. *Computational Culture, 7*. Retrieved from http://computationalculture.net/the-patterning-of-finance-security/

Distelmeyer, J. (2019). From object to process: Interface politics of networked computerization. *Artnodes, 24*, 83–90. https://doi.org/10.25969/mediarep/12920

Duguay, S. (2017). Dressing up Tinderella: Interrogating authenticity claims on the mobile dating app Tinder. *Information, Communication & Society, 20*(3), 351–367. https://doi.org/10.1080/1369118X.2016.1168471

Duguay, S. (2019). You can't use this app for that: Exploring off-label use through an investigation of Tinder. *The Information Society*, *36*(1), 30–42. https://doi.org/10.1080/01972243.2019.1685036

Duguay, S., & Gold-Apel, H. (2023). Stumbling blocks and alternative paths: Reconsidering the walkthrough method for analyzing apps. *Social Media + Society, 9*(1), 1–10. https://doi.org/10.1177/20563051231158822

Ernst, W. (2016). *Chronopoetics: The temporal being and operativity of technological media*. Lanham, MD: Rowman & Littlefield.

Hagen, S. H. (2024). *Reactionary rhythm: Quali-quantitative studies on 4chan/pol* (Doctoral dissertation, University of Amsterdam). Retrieved from http://hdl.handle.net/11245.1/387bd6be-268c-4d47-b031-3cf117215f8d

Hansen, M. B. N. (2015). *Feed-forward: On the future of twenty-first-century media*. Chicago, IL: University of Chicago Press.

Jacobsen, B. N. (2024). When is the right time to remember? Social media memories, temporality and the kairologic. *New Media & Society, 26*(5), 2872–2888. https://doi.org/10.1177/14614448221096768

Lai, S. S., & Flensburg, S. (2021). Invasive species of the app ecosystem: Exploring the political economy of mobile communication. *International Journal of Communication, 15*, 2301–2318.

Langlois, G. (2014). *Meaning in the age of social media*. New York, NY: Palgrave Macmillan.

Light, B., Burgess, J., & Duguay, S. (2018). The walkthrough method: An approach to the study of apps. *New Media & Society, 20*(3), 881–900. https://doi.org/10.1177/1461444816675438

Lohmeier, C., Kaun, A., & Pentzold, C. (2020). Making time in digital societies: Considering the interplay of media, data, and temporalities—An introduction to the special issue. *New Media & Society, 22*(9), 1521–1527. https://doi.org/10.1177/1461444820913555

https://doi.org/10.65476/bd4srf52

Lomborg, S., Sick, K., Flensburg, S., & Lai, S. S. (2024). Monitoring infrastructural power: Methodological challenges in studying mobile infrastructures for datafication. *Internet Policy Review, 13*(2), 1–28. https://doi.org/10.14763/2024.2.1763

MacLeod, C., & McArthur, V. (2019). The construction of gender in dating apps: An interface analysis of Tinder and Bumble. *Feminist Media Studies, 19*(6), 822–840. https://doi.org/10.1080/14680777.2018.1494618

Manovich, L. (2013). *Software takes command: Extending the language of new media*. London, UK: Bloomsbury.

Mejias, U. A., & Couldry, N. (2019). Datafication. *Internet Policy Review*, *8*(4). https://doi.org/10.14763/2019.4.1428

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

Pybus, J., & Coté, M. (2024). Super SDKs: Tracking personal data and platform monopolies in the mobile. *Big Data & Society, 11*(1). https://doi.org/10.1177/20539517241231270

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

Schüll, N. D. (2012). *Addiction by design*. Princeton, NJ: Princeton university press.

Soderman, B. (2021). *Against flow: Video games and the flowing subject*. Cambridge, MA: MIT Press.

Striphas, T. (2015). Algorithmic culture. *European Journal of Cultural Studies, 18*(4–5), 395–412. https://doi.org/10.1177/1367549415577392

Weltevrede, E., & Jansen, F. (2019). Infrastructures of intimate data: Mapping the inbound and outbound data flows of dating apps. *Computational Culture, 7*, 1–57. Retrieved from http://computationalculture.net/infrastructures-of-intimate-data-mapping-the-inbound-and-outbound-data-flows-of-dating-apps/

Wilken, R., Burgess, J., & Albury, K. (2019). Dating apps and data markets: A political economy of communication approach. *Computational Culture, 7*, 1–26. Retrieved from http://computationalculture.net/dating-apps-and-data-markets-a-political-economy-of-communication-approach/