SM+S
social media + society

Article

Social Media + Society
January-March 2022: 1–13
© The Author(s) 2022
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/20563051221077032
journals.sagepub.com/home/sms
SAGE

Bridging the Open Web and APIs: Alternative Social Media Alongside the Corporate Web

Jack Jamieson[1,2], Naomi Yamashita[1], and Rhonda McEwen[2]

Abstract

Concentrations of power over the internet among a small number of corporate platforms have motivated attempts to build alternative social media. Using the contemporary internet routinely involves relying on a small number of dominant corporate platforms. In reaction against this centralization of power, there are many attempts to build alternative Web technologies that reconfigure the internet's power structures and enact their own values. However, given the entrenchment of large corporate platforms, this typically involves co-existing with rather than replacing them, at least in the present. Accordingly, it is important to investigate challenges arising when alternative social media operate alongside and even within the systems to which they propose an alternative. We investigate this through an empirical study of the IndieWeb, a community of personal websites with social networking features including syndication to and from corporate platforms. Using GitHub data, we study the development of a tool for this syndication called Bridgy, focusing on its relationship with the Facebook API. By identifying breakdowns in this relationship, we identify the following challenges: translating differing logics between the open Web and APIs, occasional ambiguity in Facebook's presentation of privacy settings, and ongoing precarity due to API updates. Our analysis illustrates the reality of maintaining alternative technical systems as part of present-day infrastructures and generates insights for building socially empowering technologies for the future.

Keywords

alternative social media, platforms, application programming interfaces, Facebook

Introduction

Recent concerns about the dominance of social media giants highlight a need to consider alternative ways of engaging with the social Web (Gehl, 2015; Halpin, 2018). In large part, objections to corporate social media have advocated opting-out, as demonstrated by hashtag campaigns like *#deletefacebook* and efforts to build alternatives to large corporate platforms such as Mastodon, Secure Scuttlebutt, Diaspora*, and others. These projects' designs are generally motivated by commitments to social values, which are typically in conflict with the values articulated in existing systems (Kahle, 2015). While there are a variety of motivations for people to leave corporate social media (e.g., Zannettou et al., 2018), attracting users away from ubiquitous corporate platforms is a significant challenge. In light of this, some technologists have pursued a more moderate approach, attempting to reconfigure rather than abandon relationships with corporate social media. This raises a question about how designers can pursue their goals while navigating infrastructures characterized by competing values.

Significantly, these approaches confront the messy social and political-economic structures of present internet technologies. Bell and Dourish (2007) observe that the tendency to view technologies as part of a "proximate future" allows technologists to "absolve themselves for responsibilities for the present" and to "assume that certain problems will simply disappear of their own accord" (p. 134). The reality is far messier. For example, cryptocurrencies that offer a radically decentralized vision of future financial systems are presently subject to governmental pressures (Yu & Koh Ping, 2021),

[1]NTT Communication Science Laboratories, Japan
[2]University of Toronto Mississauga, Canada

Corresponding Author:
Jack Jamieson, NTT Communication Science Laboratories, 2-4, Hikaridai, Seika-cho, "Keihanna Science City" Kyoto, Japan 619-0237.
Email: jack@jackjamieson.net

traded by major wall-street firms (Leising, 2021), and have significant environmental impacts (Stoll et al., 2019). Similarly, new Web technologies are always embedded in complex and heterogeneous sociotechnical infrastructures that project power relationships onto the work of building them. Understanding the reality of building in the present is crucial for achieving realistic possibilities for the future, and so accounting for these power relationships is one of our core motivations.

This article investigates frictions and compromises that arise when alternative social media operate alongside and even within the infrastructures to which they claim to provide an alternative. To understand this phenomenon, we study the IndieWeb, a "people focused alternative to the 'corporate web'" in which individuals create and operate personal Web sites that function as their primary online identity (2020). IndieWeb has developed standards, software, and practices that support peer-to-peer communication among personal Web sites while optionally maintaining connections with large platforms. These connections are achieved through a model called *POSSE* (Publish on Own Site, Syndicate Elsewhere) in which content from personal Web sites is cross-posted to Facebook, Twitter, and other platforms, and then comments, likes, and other responses are retrieved back. This involves interaction between IndieWeb sites and corporate platforms through their Application Programming Interfaces (APIs), which clearly demonstrates the type of relationship with which this study is concerned. Moreover, open-source code and rigorous documentation of problem-solving are significant features of IndieWeb's culture. This makes it possible to observe development and maintenance over time, including how developers respond to emergent obstacles.

We leverage GitHub data to identify points of friction and breakdown in the relationship between IndieWeb and corporate APIs. In this study, we particularly focus on "Bridgy," a prominent IndieWeb software for POSSE syndication, and Facebook. We use logged GitHub issues to identify bugs, breakdowns, and other challenges, and supplemented this analysis with an interview with Bridgy's lead developer. Our analysis is structured around the following questions: (1) How might IndieWeb's goals be challenged by its reliance on corporate APIs? (2) If problems arise, how are they addressed? We identify three major recurring challenges: competing logics for mapping objects (e.g., posts, comments, likes) between IndieWeb sites and Facebook's API; occasional ambiguity about privacy on Facebook; and ongoing precarity due to API updates. We highlight the use of heuristics, workarounds, and ongoing infrastructuring work to navigate these challenges.

This study contributes to research about building technologies to serve social and ethical outcomes related to designers' values. Prior research has investigated how design decisions impact social and ethical outcomes of technologies (Friedman & Hendry, 2019; Shilton et al., 2013). Much of this research has focused on early stages of design, although recently, there is growing attention to values in later stages of development and maintenance (e.g., Houston et al., 2016; Whittle, 2019). We contribute to this emerging body of research with particular attention to power relationships between new Web technologies and established corporate platforms, especially how these relationships impact the work of building and maintaining alternatives. In addition, researchers have used GitHub data to study social structures in software development (Cheng & Guo, 2019; Strzalkowski et al., 2019), and GitHub issues in particular to understand community governance around codes of conduct (Li et al., 2021). However, to our knowledge, ours is the first study to use GitHub issues to examine how interoperability among software can challenge the ongoing achievement of desired social and ethical outcomes.

Given the ubiquity of large corporate internet platforms, emerging alternatives are likely to operate alongside rather than replacing them, at least for the time being. Accordingly, it is important to consider how co-existence with mainstream technologies shapes the conditions in which alternatives are built. Studying this sort of intervention illuminates the heterogeneous and messy present context in which alternative Web technologies are built. Our analysis reveals power dynamics, competing logics, and potential ways forward for alternative Web technologies that must contend with and extend from present-day infrastructures. The findings of this study will benefit designers of alternative systems, as well as app-developers, researchers, and others who rely on corporate platform APIs. Particularly, by mapping obstacles on a longitudinal basis, this research contributes to knowledge about planning for challenges that may emerge in later stages of developing and maintaining Web technologies.

Background

Social Media as Alternative Media

Concepts about "alternative media" provide vital scaffolding for this analysis. Couldry and Curran (2003) define alternative media as "media production that challenges, at least implicitly, actual concentrations of media power, whatever form those concentrations may take in different locations" (p. 7). Atton (2002) similarly explains that alternative media are "crucially about offering the means for democratic communication to people who are normally excluded from media production" (p. 4). By these definitions, alternative media are not defined only by their content, but more significantly by the processes of their production and distribution, and particularly how these reshape social and power relations around media.

When contrasted with previous forms of corporate media (e.g., broadcast and print media), Web 2.0 appears to match Atton's definition by providing tools for everyday people to engage in online media production and distribution. And yet, Gehl (2015) observed,

> This leaves alternative media theory in a double-bind: social media allow for people to be producers, certainly more so than traditional media, but they are owned by for-profit firms who can be hostile to alternative ideas, discourses, and organizing—especially when those practices challenge corporate hegemony. (p. 1)

Alternative social media and Web technologies, such as Mastodon, IndieWeb, Disapora*, Secure Scuttlebutt and others, demonstrate yet new arrangements of media power, challenging the hegemony of corporate social media. Gehl (2015) argues that these emerging technologies merit scholarly attention "because they give us new ways to think about media, media infrastructures, and mediated social interactions" (p. 8). This motivates our decision to study alternative social media with consideration for power relationships. Specifically, we align this work with Cohen's (2012) argument that "if we follow Atton in understanding alternative media through organizational processes and resulting social relations, we need to consider labour" (p. 208). To that end, we focus on the work involved in building and maintaining alternative social media technologies, including the extent to which relationships with existing platforms influence what is possible and how much work it takes to achieve.

Values, Infrastructures, and Platforms

Commitments to social values are a significant feature of alternative social media (Gehl, 2015) and of emerging decentralized Web technologies more generally. For example, internet Archive founder Brewster Kahle called for builders of decentralized Web technologies to "bake our values into the code" (Kahle, 2015). Research about values and design can help us understand how this could be meaningfully pursued.

There has been substantial research in Human-Computer Interaction (HCI) and related fields about addressing values during design processes, such as Value-Sensitive Design (VSD) (Friedman & Hendry, 2019), reflective design (Sengers et al., 2005), and values at play (Flanagan et al., 2005). These approaches are focused on building technologies that enhance "human well-being, human dignity, justice, welfare, and human rights" (Friedman & Kahn, 2003, p. 1186). One lesson from this research is that "baking" values into a design should not be interpreted as somehow fixing them as mandatory outcomes of a technology. Most values and design research adopts an interactional perspective, affirming that "whereas the features or properties that people design into technologies more readily support certain values and hinder others, the technology's actual use depends on the goals of the people interacting with it" (Friedman & Kahn, 2003, p. 1179). In addition, some values may be latent except in specific circumstances, when they shift from potential to performed values (Shilton et al., 2013). Although the majority of values and design research focuses on early stages of design, recent work has begun extending these theories to "more technical stages of development" involved in software engineering (Whittle, 2019). Focusing on values in these stages can identify moments where values are shaped by interactions among various software, standards, developers, documentation resources, dependencies, and other entities. This is especially pertinent for understanding large sociotechnical systems such as the Web, where an abundance of such relationships leads to instability and breakdowns. To account for this, we draw from scholarship about social and ethical features of internet infrastructures and platforms.

Internet infrastructures are defined by complexity, involving "a baffling network of relationships producing significant outcomes that no single actor seems particularly able to foresee" (Sandvig, 2013, p. 89). Understanding these relationships is made more difficult by the fact that infrastructures tend to recede into the background when they are operating smoothly (Star, 1999). Breakdowns—moments where a system fails or is unsuitable to meet particular needs—create opportunities for infrastructural inversions (Star, 1999), figure/ground shifts that draw attention to background infrastructures. Scholars can capitalize on infrastructural inversions to uncover tacit "infrastructuring" labor, which involves "in-situ design work of tailoring and configuring the infrastructure" to accommodate whatever activities are stifled by the breakdown (Pipek & Wulf, 2009, p. 458). During these moments, a technology's values may be reinterpreted or reshaped, and thus maintenance, use, and later stages of development are important to address when studying values and design.

As well as infrastructure studies, we draw from related work about platforms (e.g., Helmond, 2015). Similar to infrastructure studies, platform studies follow a material-technical approach to investigate "the connection between technical specifics and culture" (Bogost & Montfort, 2009, p. 5). These approaches differ in that infrastructure studies emphasizes "ubiquity, reliability, invisibility, gateways, and breakdown" while platform studies highlight "programmability, affordances and constraints, connection of heterogeneous actors, and accessibility of data and logic through application programming interfaces (APIs)" (Plantin et al., 2016, p. 2). Given that corporate platforms are the mainstream against which alternative Web technologies cast themselves, the particulars of platforms are a vital consideration for this research. Notably, platform business models are based on occupying an intermediary position, bringing together users, content producers, advertisers, and other parties (Gillespie, 2010). The reliance of third-party developers upon platforms as part of this arrangement is a major focus of this research. Thus, a platform perspective helps unpack power relationships, such as how platform APIs support and constrain flows of data to serve platform goals.

IndieWeb

Our research is focused on the nexus of the IndieWeb and platform APIs, which it uses as a distribution mechanism. IndieWeb's website describes it as "a community of individual personal websites, connected by simple standards, based on the principles of owning your domain, using it as your primary identity, to publish on your own site (optionally syndicate elsewhere), and own your data" (IndieWeb.org, 2021). IndieWeb's core motivations are to increase individual's autonomy and empowerment online (Çelik, 2016). Individuals are to be empowered by both owning and controlling their Web content, rather than relying on platforms for hosting, as well as by connecting to "all services, not just one" (IndieWeb.org, 2020). IndieWeb community members commonly post first to their personal website, and then syndicate to a variety of other platforms.

IndieWeb's community has produced several standards for adding social features to personal websites. Two of the most salient for this paper are "Microformats 2" (MF2) and "Webmention." MF2 is a semantic markup standard with which one can add machine-readable context to HTML, such as by specifying the title, author, publication date, type of post (e.g., article, like, reply, photo), or other properties of some Web content. Webmention is a standard for notifying when one webpage links to another.

In combination, these standards allow interactions such as replies, likes, shares, and other familiar social media actions to be communicated among individual websites. As described in the Webmention spec (Parecki, 2017), if Bob uses his website to write a reply to Alice, he can send a Webmention with the following content:

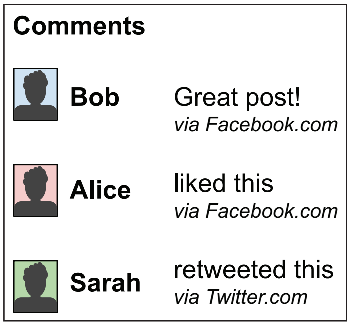
```
source=http://www.bob.example/post-by-bob

target=http://www.alice.example/post-by-alice
```

"Source" indicates the URL sending a Webmention and "target" indicates the recipient. If Alice chooses, her site may display Bob's reply, in which case it will parse the content from the source URL. For example, if Bob has added MF2 markup indicating the post is a "like," Alice's site could display, "Bob liked this post." Webmention and MF2 are designed to be generalizable, so Bob and Alice's websites can communicate even even if they have different structures or use different software. These standards are core mechanisms for communication among IndieWeb sites, and form an important part of IndieWeb's systems for syndicating to and from corporate platforms, discussed presently.

Syndicating to Platforms With Bridgy

One of IndieWeb's defining features is its emphasis on maintaining connections to platforms through an approach called POSSE (Publish on Own Site, Syndicate Elsewhere). Although IndieWeb's premise is that people use their own websites as their main online presence, they do not necessarily avoid corporate social media altogether. IndieWeb users often syndicate from their personal Web sites to Facebook, Twitter, and other platforms, and then retrieve comments, likes, and other responses back to the original post. IndieWeb co-founder Tantek Çelik described two advantages to this approach: (1) It allows one to keep in touch with friends or family across social media, (2) by aggregating interactions from many platforms in one place, one "has an experience on his site that's better than any silo" [i.e., corporate Web platform] (Çelik, 2014).



Comments

Bob Great post! via Facebook.com

Alice liked this via Facebook.com

Sarah retweeted this via Twitter.com

Figure 1. After syndicating content from one's website to various social media platforms, Bridgy can be used to "backfeed" likes, retweets, and other responses. This is typically used to display responses from across social media alongside the original post.

The most popular tool for POSSE is a Web service called Bridgy, which "pulls comments, likes, and reshares on social networks back to your Web site. You can also use it to post to social networks—or comment, like, reshare, or even RSVP—from your own web site" (Bridgy, 2020). Figure 1 shows an example of how responses from social media platforms can be displayed on the original post. Like many IndieWeb tools, Bridgy is free and generates no revenue. Bridgy works by translating between popular platform APIs and IndieWeb sites that support Webmentions and MF2. In contrast to IndieWeb's generalizable standards, APIs vary across platforms, each allowing access to different types of data and requiring distinct methods to facilitate that access. Thus, when Bridgy connects to various platforms, its features vary according to the constraints of individual APIs. Figure 2 represents the overall data flow when syndicating content from an IndieWeb site to a platform such as Facebook, and then "backfeeding" likes, comments, and other responses from the platform to the original post. Importantly, communication between the IndieWeb site and Bridgy uses Webmentions, which require the "source" content to be posted at a publicly accessible URL.[1] Thus, Bridgy does not support syndication of private content.

Bridgy is one of the most widely used pieces of IndieWeb software. To put Bridgy's scope in context, when IndieWeb celebrated that one million Webmentions had been sent in the wild, 960,778 had been sent using Bridgy (Barrett, 2018a).

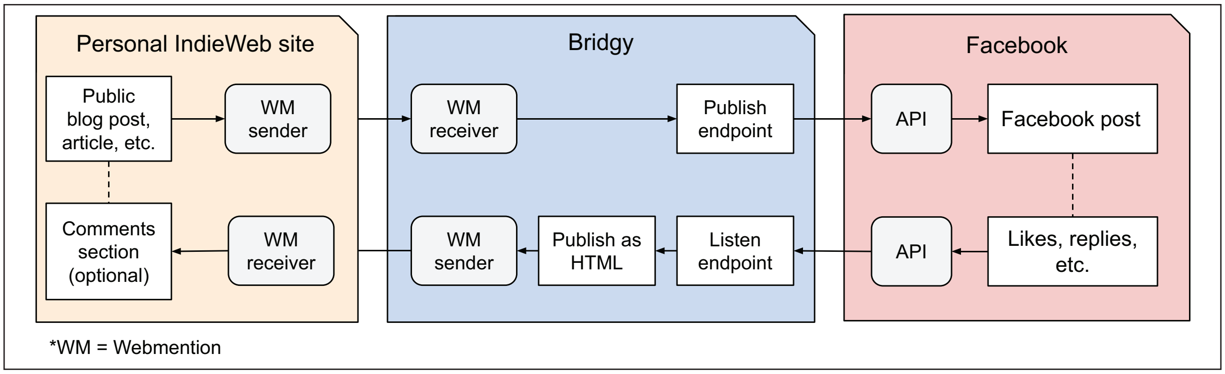
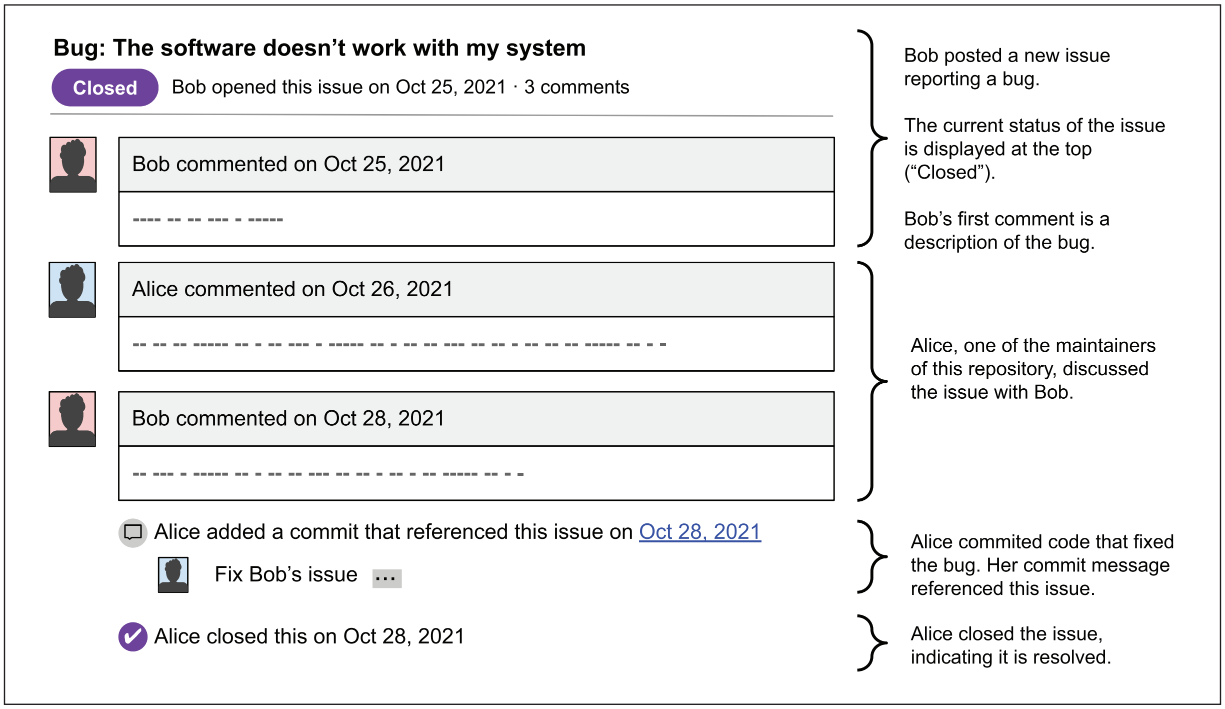


Figure 2. Data flow between IndieWeb site, Bridgy, and Facebook. Top row: Syndicating from an IndieWeb site to Facebook. Bottom row: "Backfeeding" responses from Facebook to an IndieWeb site.



Bug: The software doesn't work with my system

Closed Bob opened this issue on Oct 25, 2021 · 3 comments

Bob commented on Oct 25, 2021

---- -- -- --- - -----

Alice commented on Oct 26, 2021

-- -- -- ------ -- - - -- --- - ------ -- - -- -- --- -- -- - -- -- -- ------ -- - -

Bob commented on Oct 28, 2021

-- ---- - ------ -- - -- -- --- -- -- - -- - -- ----- -- - -

Alice added a commit that referenced this issue on Oct 28, 2021

Fix Bob's issue …

✓ Alice closed this on Oct 28, 2021

Bob posted a new issue reporting a bug.

The current status of the issue is displayed at the top ("Closed").

Bob's first comment is a description of the bug.

Alice, one of the maintainers of this repository, discussed the issue with Bob.

Alice commited code that fixed the bug. Her commit message referenced this issue.

Alice closed the issue, indicating it is resolved.

Figure 3. Basic structure of a GitHub issue. Issues are discussion threads about software bugs or features, which can contain references to specific code commits.

Moreover, Bridgy has been recognized as influential for opening up IndieWeb beyond its early adopters, since it was the first broadly accessible tool for connecting IndieWeb to corporate social media. Accordingly, Bridgy is a valuable site through which to investigate IndieWeb's relationship with corporate platforms and therein to map out considerations for co-existence among alternative and mainstream Web technologies.

Method

Like most IndieWeb projects, Bridgy is open source and hosted on GitHub, a platform for software code and other version-controlled repositories. GitHub repositories include a detailed history of revisions to source code and other documents, making it possible to identify how a project has changed over time. Individual changes are published as "commits" and discussion threads about new features and bugs are archived as "issues." Any GitHub user can post an issue to a repository or participate in discussions about existing issues. When people commit code to the repository, they can reference the issue in a "commit message," in which case the issue thread will include a link to that commit. Figure 3 shows a rudimentary example of how issues are structured.

Issues provide an apt entry-point for investigating Bridgy's connections to other systems. First, they highlight breakdowns, since most literally describe errors, bugs, and other problems. Second, they present traces of the work that was conducted to resolve these breakdowns. In many cases, these discussions reference specific commits in which an issue was addressed or resolved, making it possible to observe deliberation and troubleshooting in developer discussions, and then to identify ways that issues were addressed through code.

We downloaded copies of Bridgy's issues for analysis. Between the earliest item (4 January 2014) and the time of downloading (5 March 2018), there were 799 issues. We then performed a keyword search to get a general sense of how many issues pertained to each platform, with the following results: Twitter (*N*=316), Facebook (*N*=278), Google (*N*=165), Instagram (*N*=99), Flickr (*N*=50), and Blogger (*N*=32).[2] This is not an exact measure because the presence of a keyword does not guarantee that the issue actually pertains to that platform, and some keywords are ambiguous ("Google" could indicate the search engine, Google+, or the company generally). Nonetheless, this preliminary query contributes a rough sketch of where Bridgy has had the most issues in relation to corporate platforms.

We focused on issues involving Facebook for two reasons. First, they represent a significant proportion of Bridgy's total issues. Second, Facebook's large scope means these issues demonstrate many different types of interactions and potential problems. In an interview, Bridgy's creator remarked that, "Any feature in any silo, Facebook has it too. And then they also have ten or a hundred thousand features that silo doesn't. There aren't many concerns that you don't see in Facebook."

A first pass reading of the 278 issues that included the term "Facebook" revealed that 147 of them described some sort of problem or feature request related to Facebook's API. We then used open coding to organize the issues into categories describing types of problems that were most prominent. Where possible, we followed links to code commits that were referenced in the issue discussions, which assisted in identifying how Bridgy's developers responded to different types of challenges.

While studying GitHub data, we were mindful for potential pitfalls presented by Kalliamvakou et al. (2016), two of which are relevant for this study. First, like many projects, Bridgy is not contained wholly within a single repository, instead relying on pieces from other repositories to perform some functions. Studying a single repository without considering its dependencies can lead to an incomplete understanding. A benefit of our approach is that issues posted to the main Bridgy repository often reference commits to its dependencies. Therefore, it was possible to start with Bridgy and expand outward to its dependencies, avoiding a potential blind spot.

Second, GitHub projects almost always involve development and discussion occurring outside of the GitHub platform. Accordingly, this study is supplemented by ongoing participant-observation of IndieWeb's developer community and a semi-structured interview with Bridgy's creator and lead developer, Ryan Barrett. This interview served to verify our assessment of the technical challenges and approaches as discovered through analysis of its GitHub repository, as well as to develop a richer understanding of how the challenges were addressed.

Findings

Like most IndieWeb projects, Bridgy is largely built by one person. At the time of analysis, approximately 92% (*N*=2401) of Bridgy's 2606 commits were made by Bridgy's creator and lead developer, Ryan Barrett. However, Bridgy brings together a variety of IndieWeb community members. Eleven people have made commits to Bridgy's code, and 123 people have posted or commented on issues. Three types of problems emerged as the most substantial and recurring causes of breakdowns described in Bridgy issues: Mapping between URLs and API IDs, occasionally ambiguous privacy statuses, and ongoing precarity due to API updates. We summarize these problems here as well as developers' responses.

Mapping Between URLs and API IDs

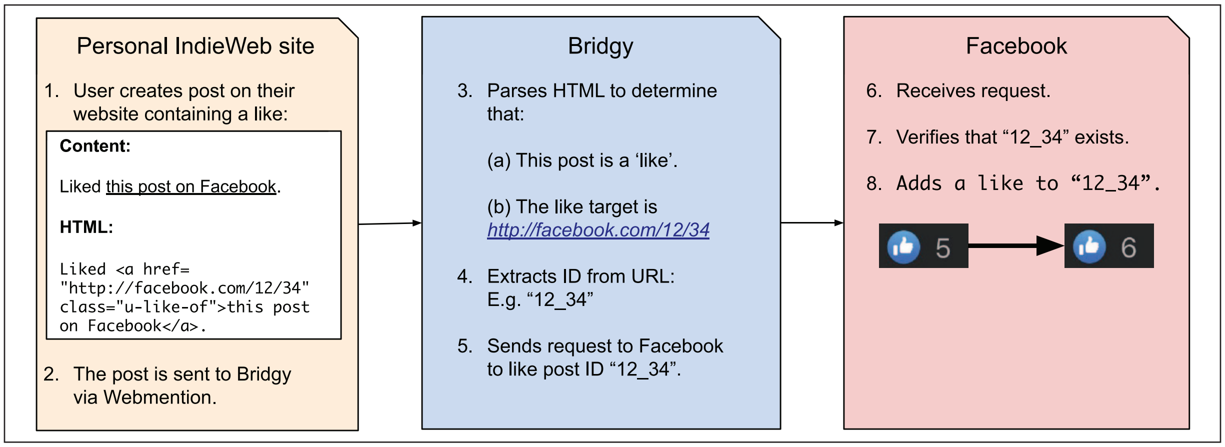
When syndicating between personal Web sites and Facebook, Bridgy spans a threshold between the open Web and platform APIs. The most prominent challenge in Bridgy's development has been translating between different ways of addressing an object across this boundary. Objects on the open Web are addressed using URLs, such as *http://facebook.com/{user-id}/{object-id}*, whereas the same object could be identified within Facebook's API using an ID in a format such as *{user-id}_{object-id}*.

In some instances, it is straightforward to translate between these formats. In the example above, one can map between these identifiers using {*user-id*} and {*object-id*}. However, this mapping is often quite difficult for a variety of reasons. Notably, Facebook IDs come in multiple formats, represented in Table 1.[3] Bridgy's developers struggled to predict which format is required in different cases, and Bridgy's code resorts to trial and error to find the correct format.

Mapping between URLs and API IDs became more difficult in 2014, when Facebook released version 2.0 of its API. This update limited the amount of data third-party developers could access. In 2018, when Facebook CEO Mark Zuckerberg (2018) testified before the U.S. House of Representatives Committee on Energy and Commerce about Cambridge Analytica's collection of Facebook users' data, he asserted that this update "makes it so a developer today can't do what Kogan [the developer who shared data with Cambridge Analytica] did years ago." One of the changes was the introduction of app-scoped user IDs, which mean that each third-party app is given a different ID for the same user. This improves Facebook users' privacy and security by frustrating efforts to combine data collected by multiple apps. However, this also complicates Bridgy's efforts to map

Table 1. Types of Formats for Facebook IDs. Adapted Comment From Bridgy's Code.

Format	Example
"Simple number, usually a user or post"	12
"Two numbers with underscore, usually POST_COMMENT or USER_POST"	12_34
"Three numbers with underscores, USER_POST_COMMENT"	12_34_56
"Three numbers with colons, USER:POST:SHARD"	12:34:63
"Two numbers with colon, POST:SHARD"	12:34
"Four numbers with colons/underscore, USER:POST:SHARD_COMMENT"	12:34:56_56
"Five numbers with colons/underscore, USER:EVENT: UNKNOWN_UNKNOWN. Not currently supported!"	111599105530674:195181727490727: 10102446236688861: 10205257726909910_195198790822354



1. User creates post on their website containing a like:

Content:

Liked this post on Facebook.

HTML:

```
Liked <a href=
"http://facebook.com/12/34"
class="u-like-of">this post
on Facebook</a>.
```

2. The post is sent to Bridgy via Webmention.

3. Parses HTML to determine that:

(a) This post is a 'like'.

(b) The like target is http://facebook.com/12/34

4. Extracts ID from URL: E.g. "12_34"

5. Sends request to Facebook to like post ID "12_34".

6. Receives request.

7. Verifies that "12_34" exists.

8. Adds a like to "12_34".

Figure 4. Basic process of posting a like from one's personal website using Bridgy, which is no longer possible since the release of Facebook's API version 2.0.

between a URL on Facebook.com and the corresponding object in Facebook's API.

This led a notable reduction in Bridgy's capabilities. Previously, Bridgy could be used to like a Facebook post from one's own website, following the process in Figure 4. This is no longer possible because URLs to view a Facebook post in one's browser refer to users by a "global" numerical ID or by their username (which Bridgy can use to find their global ID). By design, it is not possible to map a global ID to an app-scoped ID, which is required to send or access data in Facebook's API. As a result, the switch to app-scoped IDs resulted in Bridgy dropping this feature.

In most other cases, Bridgy has been successful at mapping between URLs and API IDs, albeit with considerable effort. From Facebook's perspective, IDs are to be taken as opaque objects, and Bridgy's attempts to decode them are not supported. Facebook support has advised Bridgy's developers, "Please treat IDs as unique strings, they are not meant to be broken down and used" (Facebook for Developers, 2015b). However, it is only when broken down and used that these IDs can be translated in URLs:

> There's no consistent way, either through the API or through an algorithm you implement yourself, to say, "Here is a Facebook post, what is its permalink on Facebook.com?" We have to guess at that with a surprising number of heuristics. Not ideal. So that mapping back and forth between the Web and data inside Facebook has been the single biggest question. (Interview with Ryan Barrett)

Bridging the open Web with Facebook's API necessitates reconciling differences across both sides. Ultimately, IDs used to reference Facebook content act as boundary objects, allowing different groups to work together in the absence of consensus by "tacking" back and forth between an object's general form, which is meaningful across communities, and specific forms of the object tailored to meet local needs (Star, 2010). Bridgy and Facebook share a general understanding of IDs as a means of referring to objects, but differ in their local expectations of how IDs should be used. As a result, Bridgy uses heuristics, trial and error, and similar methods to fit the shared general understanding of IDs with its local requirements.

Privacy From Front-End to Back-End

There are two main mechanisms for users to control who accesses their data on Facebook. First, Facebook's privacy

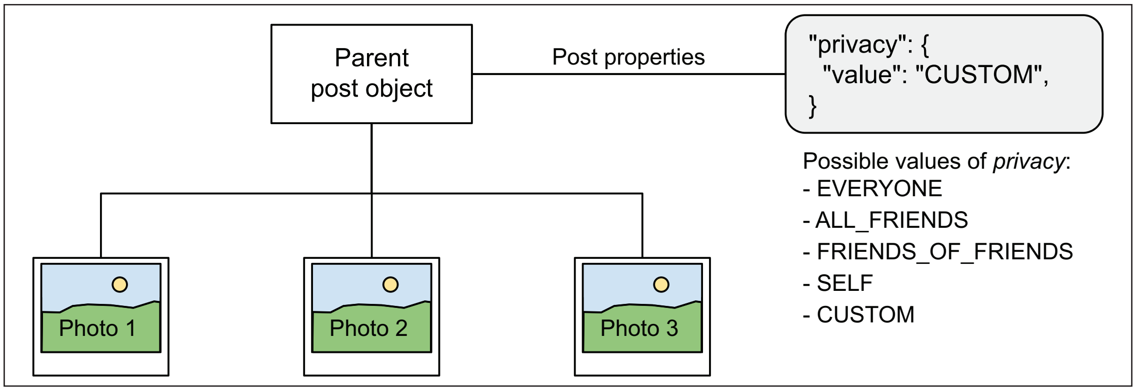


Figure 5. In Facebook's API, photo objects themselves do not have a privacy field, so Bridgy checks the privacy field of their parent post.

settings can specify an audience that can view their content using Facebook's website or app. Options include "Public," "Friends," "Friends except [...]," "Specific friends," and "Only me." Second, whenever a third-party app such as Bridgy wants to access Facebook data, users must grant it permission. For example, to access posts and photos, a third-party app will display a notice that the app will receive "your timeline posts and photos"—If the user agrees, the app is granted the *user_posts* and *user_photos* permissions to access that data in Facebook's API.

These two privacy mechanisms are independent, and there is no way to grant a third-party developer access to public photos, but not to photos intended for friends only. As a result, it has been up to third-party developers to manage their treatment of posts with different audiences.

As shown in Figure 1, the data flow when 'backfeeding' responses from Facebook to an IndieWeb site involves publishing the response as a publicly accessible HTML page Consequently, when accessing content in Facebook's API, Bridgy checks its privacy status and ignores any content not marked explicitly as 'public.' Barrett commented on this process:

> That is non-trivial to determine for a given object in the Facebook API, is it public? Usually, 90% of the time, it's straightforward. Another 9% it takes some work, but you can figure it out, again depending on the type. It's like 1% or maybe 0.1% where you actually can't tell [. . .] If you go look at the [User Interface] in Facebook, you can usually tell. But programmatically you can't. When that happens, I have to err on the side of not doing anything with it. (Interview with Ryan Barrett)

The most striking example of this difficulty can occur with photos. In Facebook's API, photos themselves do not possess a privacy status. Instead, each photo is part of a parent post and/or album, which contains a privacy field indicating the intended audience, as shown in Figure 5. Therefore, determining the privacy status for a photo in Facebook's requires one to check this parent.

When someone posts multiple photos to their Facebook timeline in a short period of time, Facebook creates a parent post representing them as a group even if the individual photos have distinct privacy settings. However, in Facebook's API, the parent post object may have a privacy status of "CUSTOM." In a GitHub issue, Barrett explained that Facebook's documentation "[does not] say anything about what CUSTOM with no details means."[4] As a result, there are rare cases where it is not possible for Bridgy to determine the privacy status of an object in the API, even if it would be apparent through Facebook's user interface.

When this was reported as a potential bug to Facebook's developer support, the response indicated that Bridgy's method of constructing the photo object's ID resulted in accessing the photo through an unusual path where the privacy status could be unidentifiable. Specifically, following the structures in Table 1, Bridgy's navigation of Facebook's API included parsing an ID string like "USER_POST_COMMENT" into its three parts, then reassembling those parts to access related API objects. However, the support representative indicated that one should *never* have to construct object IDs in this way (Facebook for Developers, 2015a). This demonstrates that Bridgy's usage was beyond Facebook's expectations and highlights how unconventional efforts to repurpose platforms can surface hidden features or limitations of their systems.

The decision to ignore data that is not explicitly marked as public has led to cases where Bridgy fails to process content that is intended to be public if the privacy setting in Facebook's API is unclear. This has been interpreted by multiple users as a bug on Bridgy's part, since the data is marked as public in Facebook's user interface and yet was ignored by Bridgy.

This example helps position the importance of privacy in Bridgy's design. IndieWeb is generally less concerned with privacy than some other alternative social media, evidenced by its practice of syndicating content to corporate platforms and the fact that IndieWeb sites are almost always publicly accessible. However, even though Bridgy does little to enhance individuals' privacy, when a threat emerged to existing privacy expectations (that a private Facebook photo should not be made public), addressing it took priority over other aspects of Bridgy's design.

Precarity and API Updates

APIs can change quickly and unpredictably. During this study, Facebook issued security and privacy API updates in light of the Cambridge Analytica scandal, which removed the ability for third-party apps to publish content to one's Facebook account (Archibong, 2018) and to retrieve data from anyone who had not directly authorized the app to access their account. This meant Bridgy could no longer access responses from people who were not also Bridgy users, which constituted "probably 98-99%" of interactions Bridgy had been backfeeding from Facebook (R. Barrett, personal communication, January 31, 2019). Since this closed off the vast majority of Bridgy's Facebook's functionality, Bridgy dropped support for Facebook altogether (Barrett, 2018b).

During our interview, Barrett was quick to assert that problems he encountered with Facebook's API were not a result of malice. Rather, "in many ways what Bridgy is doing is not at all what Facebook expects the average Facebook app to do." Furthermore, he asserted that an app like Bridgy is simply too small to be a concern for Facebook, citing his past experiences as a senior engineer at Google:

> I have seen some it at the inside of Google. For things that aren't security breaches—for apps that are just doing funny things that may or may not be against your TOS [terms of service]—if they're small enough you don't care.

Thus, it is unlikely that anyone at Facebook was specifically motivated to restrict services like Bridgy. Instead, Facebook's attitude toward Bridgy could be characterized as indifference.

This indifference creates opportunities for experimentation and innovation, but as demonstrated by Facebook's API updates in 2018, also cultivates substantial risk. Another IndieWeb developer who created a commercial service with similar syndication features as Bridgy has written that reliance on platform APIs was an obstacle, especially from a business perspective, "Investors would ask about the supplier risk of being so heavily dependent on third-party APIs to provide a lot of the core value. They were right" (Werdmüller, 2018).

Although Bridgy was, for a time, successful at navigating its relationship with Facebook, the pressure of the Cambridge Analytica scandal prompted a set of restrictions in Facebook's API that Bridgy could not accommodate. While the circumstances surrounding Facebook's API restrictions were dramatic, they represent a precarity that extends to all similar dependence on platforms. Nonetheless, the purpose of POSSE in the first place is to reduce dependence upon platforms by encouraging people to post to their personal website first, rather than allowing the only copy of one's content to be on a platform. In this respect, the outcome when Bridgy has lost access to platforms demonstrates resilience of this overall approach. Specifically, losing access to Facebook's API did not result in any loss of data for users, since their posts and responses pulled from Facebook were stored on their personal websites.

Furthermore, while Facebook's API shutdown led to an overnight decrease in Bridgy accounts (Barrett, 2020), other platforms with which Bridgy supports POSSE remain functional and new platforms have been added, including Meetup, Reddit, and Mastodon. And almost 3 years after Bridgy lost support for Facebook, it returned in the form of a browser plugin (Barrett, 2021). Thus, although API changes thwart long-term stability, there are still opportunities to meaningfully repurpose and extend corporate platforms beyond their intended uses, albeit with considerable effort and tolerance for setbacks.

Discussion

Our research questions were (1) *How might IndieWeb's goals be challenged by its reliance on corporate APIs?* (2) *If problems arise, how are they addressed?* The results have highlighted the need for ongoing infrastructuring work, and a related danger of emergent ethical dilemmas (in this case, concerning privacy). Alongside this, there was persistent precarity due to the potential for API changes that modify or remove access to data.

IndieWeb's POSSE model for syndicating data between personal websites and corporate platforms is fundamentally about *artful integration*. Specifically, this model is about "the collective achievement of new, more productive interactions among devices, and more powerful integrations across devices and between devices and the settings of their use" (Suchman, 2002, p. 99). This does not indicate that Bridgy exemplifies a special and unusual type of design that relies on networks of relations, but rather that it explicates the role of these relations, which are mutually generative for all objects. This perspective of design as existing within networks of mediation is a counter to the stance of "design from nowhere," which "is closely tied to the goal of construing technical systems as commodities that can be stabilized and cut loose from the sites of their production long enough to be exported en masse to the sites of their use" (Suchman, 2002, p. 101). When Bridgy's use of Facebook's API achieved stabilization, it was a mutual accomplishment based on interactions between both actors. Yet, since Facebook was essentially indifferent to Bridgy, the burden of maintaining that accomplishment fell to Bridgy's developers, while Facebook's API changes and other inconsistencies regularly caused destabilization.

Consequently, Bridgy can never be cut loose from the site of its production. Given the extent to which it diverges from

Facebook's expected use-case, Bridgy demonstrates surprising resilience. However, platform APIs are ultimately gatekeepers, and the capacity to use them in unexpected and unintended ways is hard won. Since Web platforms generally follow a model of constant iterative development (O'Reilly, 2007), third-party developers are required to do the same. This means constantly updating their software to maintain compatibility where possible and dealing with the fallout of incompatibility where necessary. This limits the potential for building long-lasting third-party software, since this requires that developers remain active over several years of maintenance.

Earlier we defined alternative media in terms of their capacity to include people who have been marginalized from mainstream media. This makes it important to acknowledge that in the case of unpaid software development such as Bridgy's, ongoing maintenance is only possible for developers who are sufficiently committed and privileged to balance this labor against professional, family, and other obligations. This increases burdens for many groups, including women, who generally take on a greater share of unpaid social productive labor such as child care and housework (Sayer, 2005).

Researchers who rely on social media data have experienced similar challenges and offer insights that extend to our analysis. In both contexts—social media research and Bridgy's repurposing of platforms as distribution—even dramatic API restrictions can be navigated to some extent. Bruns (2019) identified four paths forward for social media researchers, which are also useful for thinking about the future of alternative Web technologies that use platform APIs.

1. *Walk away*: Bruns notes that some will stop conducting research using platform APIs. Bridgy's loss of Facebook support parallels this. Just as researchers who walk away from platform data may still work in other areas, software developers who abandon one platform may devote their time to engaging with others, as was the case with Bridgy, or to working on alternatives that do not interoperate with corporate platforms.
2. *Lobby for change*: Developers of emerging Web technologies are less likely than academics to be lobbyists for regulatory changes. Nonetheless, their work is affected by regulation. For example, Europe's General Data Protection Regulation (GDPR) includes a data portability provision that is especially relevant for software such as Bridgy, since it can be interpreted as facilitating "user-centric platforms of interrelated services"; however, other interpretations are possible (De Hert et al., 2018 p. 203) On the other hand, IndieWeb community members have debated about whether backfeeding comments from social media could itself violate the GDPR (Goldsmith, 2018; Greger, 2018). Ultimately, regulations are likely to simultaneously support and constrain emerging alternative technologies, which are often too small to warrant specific attention in regulatory design.
3. *Accommodate and acquiesce*: Bruns (2019) offers several ways in which researchers may maximize the value of increasingly restricted API data, such as by combining "disparate, incomplete, imperfect datasets" (p. 1559). Our analysis of Bridgy has highlighted the extent to which all longitudinal access to API data is likely to involve at least some accommodation of this sort. Ingenuity can push the boundaries what is possible; however, the work of maintaining applications that diverge from expected API use-cases is onerous. Reflecting on his decision to stop working on NetVizz, a research software for working with Facebook data, Rieder (2018) argued that this labor requires "substantial support from organizations capable of investing resources that may not lead to immediate deliverables." Thus, we should pay close attention to organizational and economic structures that can support alternative technologies. Recent work about community-oriented funding for alternative platforms is particularly valuable to that end (e.g., Schneider & Mannan, 2021; Scholz & Schneider, 2016).
4. *Break the rules*: The final path Bruns presents is to break the rules, such as by Web scraping HTML from platform websites. Putting aside legal and ethical questions about such an approach, the precarity-related challenges identified in this paper would be amplified when Web scraping, since platforms' HTML is easily and routinely obfuscated to frustrate machine-readability.

Bruns (2019) concludes that, at least in the short term, all of these paths will be followed by researchers. The same is true of developers who rely on platform APIs, and there are many complements among the interests of both groups.

Beyond the question of maintaining functionality at all, the ethical dilemma we observed about ambiguous privacy statuses merits special consideration. Shilton et al.'s (2013) sociotechnical dimensions of values provide language for describing the place of privacy in Bridgy's design: Salience (peripheral to central), intention (accidental to purposive), and enactment (potential to performed). The decision to only process public data reflects a performed commitment to privacy, however one that literally pushes it to the periphery. However, privacy became a vital and *central* value for resolving a situation that had not been foreseen by either Facebook's or Bridgy's developers—From Facebook's perspective, Bridgy's unconventional way of mapping objects in Facebook's API, and from Bridgy's perspective, Facebook's ambiguous documentation of privacy status when accessed through that route.

Many instances of values-oriented design take a critical approach based on challenging the status quo (e.g., Flanagan et al., 2005). In fact, this is the basis of IndieWeb's (and Bridgy's) overall effort to build alternative Web technologies that enhance individual autonomy. Yet, when it comes to privacy, Bridgy does not attempt to *enhance* individuals' privacy, but rather to avoid deteriorating Facebook's existing privacy features. We posit that design strategies for *preserving* values are particularly difficult to articulate in advance of use, since some threats to values may only emerge as a result of unforeseen relationships.

Addressing unanticipated threats to values can be served by Houston et al.'s (2016) proposed agenda to move away from a notion of values as universal properties and toward an understanding of *valuation* as "contingent, ongoing processes through which things are rendered valuable in a wider social and material context" (p. 1412). One of the consequences of this shift is an awareness that "what values are materialized and how they are made visible are deeply intertwined issues" (Houston et al., 2016, p. 1412). And thus, while some values are defined and prioritized in advance of material engagements, others may be taken for granted or otherwise invisible until a situation emerges that demands attention to a particular value. Preserving values, therefore, is less likely to involve stabilizing a design toward a particular direction, but rather identifying and responding to emergent destabilizations. To that end, research that pursues infrastructural inversions at sites of ongoing maintenance will be fruitful for extending our knowledge about the sustainability of alternative technologies, especially those that interoperate with corporate Web platforms.

Conclusion

This study has investigated the work of maintaining a system for POSSE syndication between IndieWeb sites and Facebook. Given the extent of platformization, many new systems and tools must co-exist with corporate platforms to some extent, even as they attempt to provide an alternative. Our analysis has illustrated the extent to which infrastructuring work is necessary to maintain the stabilization of such arrangements. Furthermore, we have shown that emergent situations can necessitate design interventions to preserve values that had previously been taken for granted. These are important considerations for IndieWeb's efforts to provide an alternative that operates alongside and in collaboration with corporate platforms while remaining independent of them.

In addition, this study has demonstrated that GitHub issues can be a rich analytic resource for infrastructural inversions. Norms and conventions for issues vary among different projects, but issues have several features that make them invaluable for finding connections between design goals, sociotechnical obstacles, and specific technical decisions. Scholars engaged in the long-standing and yet unfulfilled effort to bridge social and technical research will benefit by adding this sort of analysis to their toolkit.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication 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 research was supported by a Social Science and Humanities Research Council Canada Graduate Scholarship, 2015-2018.

ORCID iD

Jack Jamieson https://orcid.org/0000-0002-8444-5722

Notes

1. A small number of IndieWeb sites have implemented an extension to the Webmention spec, which allows private posts. However, private Webmentions are currently not used widely and are not supported by Bridgy.
2. At the time of this study, Bridgy had recently added compatibility for syndicating to and from GitHub. This feature was excluded from this study because it was new and lacked issues for analysis.
3. Table 1 is adapted from lines 1594–1614 at https://github.com/snarfed/granary/blob/1e2568698de6f93ce383db2631cdfddfecd9634e/granary/facebook.py
4. See https://github.com/snarfed/bridgy/issues/611/#issuecomment-174315265

References

Archibong, I. (2018, April 24). New Facebook platform product changes and policy updates. *Facebook*. https://developers.facebook.com/blog/post/2018/04/24/new-facebook-platform-product-changes-policy-updates/

Atton, C. (2002). *Alternative media*. SAGE.

Barrett, R. (2018a, January 11). 1 million webmentions. *Snarfed.org*. https://snarfed.org/1-million-webmentions

Barrett, R. (2018b, August 3). RIP Facebook for Bridgy. *Snarfed.org*. https://snarfed.org/2018-08-03_rip-facebook-for-bridgy

Barrett, R. (2020, June 27). Bridgy stats update. *Snarfed.org*. https://snarfed.org/2020-06-27_bridgy-stats-update-5

Barrett, R. (2021, February 18). Bridgy browser extension for Facebook. *Snarfed.org*. https://snarfed.org/2021-02-18_bridgy-browser-extension-for-facebook

Bell, G., & Dourish, P. (2007). Yesterday's tomorrows: Notes on ubiquitous computing's dominant vision. *Personal and Ubiquitous Computing*, *11*(2), 133–143. https://doi.org/10.1007/s00779-006-0071-x

Bogost, I., & Montfort, N. (2009). Platform studies: Frequently questioned answers. *Digital Arts and Culture*, *7*. https://escholarship.org/uc/item/01r0k9br

Bridgy. (2020). About—Bridgy. https://brid.gy/about

Bruns, A. (2019). After the "APIcalypse": Social media platforms and their fight against critical scholarly research. *Information,* *Communication & Society*, *22*(11), 1544–1566. https://doi.org/10.1080/1369118X.2019.1637447

Çelik, T. (2014, June). *Why we need the IndieWeb* [Conference session]. Personal Democracy Forum. https://www.youtube.com/watch?v=HNmKO7Gr4TE

Çelik, T. (2016, June). *State of the IndieWeb* [Conference session]. IndieWeb Summit, Portland, OR, United States. https://www.youtube.com/watch?v=7zTolqW_I2g

Cheng, J., & Guo, J. L. C. (2019). Activity-based analysis of open source software contributors: Roles and dynamics. *Arxiv:1903*. http://arxiv.org/abs/1903.05277

Cohen, N. S. (2012). From alienation to autonomy: The labour of alternative media. In K. Kozolanka, P. Mazepa, & D. Skinner (Eds.), *Alternative media in Canada* (pp. 207–225). University of British Columbia Press.

Couldry, N., & Curran, J., (Eds.). (2003). *Contesting media power: Alternative media in a networked world*. Rowman & Littlefield.

De Hert, P., Papakonstantinou, V., Malgieri, G., Beslay, L., & Sanchez, I. (2018). The right to data portability in the GDPR: Towards user-centric interoperability of digital services. *Computer Law & Security Review*, *34*(2), 193–203. https://doi.org/10.1016/j.clsr.2017.10.003

Facebook for Developers. (2015a). *Post not returning with correct privacy*. https://developers.facebook.com/bugs/983622558349347/

Facebook for Developers. (2015b). *Unusually formatted comment ids (with colons) returned by /me/posts*. https://developers.facebook.com/bugs/786903278061433/

Flanagan, M., Howe, D. C., & Nissenbaum, H. (2005). *Values at play: Design tradeoffs in socially-oriented game design* [Conference session]. SIGCHI Conference on Human Factors in Computing Systems, Portland, OR, United States. https://doi.org/10.1145/1054972.1055076

Friedman, B., & Hendry, D. (2019). *Value sensitive design: Shaping technology with moral imagination*. MIT Press.

Friedman, B., & Kahn, P. H. (2003). Human values, ethics, and design. In J. Jacko & A. Sears (Eds.), *Handbook of human-computer interaction* (pp. 1177–1201). Lawrence Erlbaum.

Gehl, R. W. (2015). The case for alternative social media. *Social Media + Society*, *1*(2), 1–12. https://doi.org/10.1177/2056305115604338

Gillespie, T. (2010). The politics of "platforms." *New Media & Society*, *12*(3), 347–364. https://doi.org/10.1177/1461444809342738

Goldsmith, D. (2018, May 5). *View from Ascraeus*. https://ascraeus.org/micro/1525556293/

Greger, S. (2018, May 2). The Indieweb privacy challenge (Webmentions, silo backfeeds, and the GDPR). *Sebastiangreger.net*. https://sebastiangreger.net/2018/05/indieweb-privacy-challenge-webmentions-backfeeds-gdpr

Halpin, H. (2018, October). Decentralizing the social web: Can blockchains solve ten years of standardization failure of the social web? INSCI 2018 - 5th International conference 'Internet Science', St. Petersburg, Russia. https://hal.inria.fr/hal-01966561

Helmond, A. (2015). The platformization of the Web: Making Web data platform ready. *Social Media + Society*, *1*(2), 1–11. https://doi.org/10.1177/2056305115603080

Houston, L., Jackson, S. J., Rosner, D. K., Ahmed, S. I., Young, M., & Kang, L. (2016). Values in repair. In: *CHI'16 proceedings of the 2016 CHI conference on human factors in computing systems* (pp. 1403–1414). Association for Computing Machinery Press. https://doi.org/10.1145/2858036.2858470

IndieWeb.org. (2020). Home. https://indieweb.org/

IndieWeb.org. (2021). IndieWeb. https://indieweb.org/IndieWeb

Kahle, B. (2015, August 11). Locking the Web open: A call for a decentralized Web. *Brewster Kahle's Blog*. http://brewster.kahle.org/2015/08/11/locking-the-web-open-a-call-for-a-distributed-web-2/

Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D. M., & Damian, D. (2016). An in-depth study of the promises and perils of mining GitHub. *Empirical Software Engineering*, *21*(5), 2035–2071. https://doi.org/10.1007/s10664-015-9393-5

Leising, M. (2021). Goldman offers new bitcoin derivatives to Wall Street investors. *Bloomberg.com*. https://www.bloomberg.com/news/articles/2021-05-06/goldman-offers-new-bitcoin-derivatives-to-wall-street-investors

Li, R., Pandurangan, P., Frluckaj, H., & Dabbish, L. (2021). Code of conduct conversations in open source software projects on Github. *Proceedings of the ACM on Human-computer Interaction*, *5*, Article 19. https://doi.org/10.1145/3449093

O'Reilly, T. (2007). What is Web 2.0: Design patterns and business models for the next generation of software. *Communications & Strategies*. https://mpra.ub.uni-muenchen.de/4580/

Parecki, A. (2017, January 12). *Webmention*. https://www.w3.org/TR/webmention/

Pipek, V., & Wulf, V. (2009). Infrastructuring: Towards an integrated perspective on the design and use of information technology. *Journal of the Association for Information Systems*, *10*, 306–322. https://doi.org/10.17705/1JAIS.00195

Plantin, J.-C., Lagoze, C., Edwards, P. N., & Sandvig, C. (2016). Infrastructure studies meet platform studies in the age of Google and Facebook. *New Media & Society*, 20, 293–310.

Rieder, B. (2018, August 11). Facebook's app review and how independent research just got a lot harder. *The Politics of Systems: Thoughts on Software, Power, and Digital Method*. http://thepoliticsofsystems.net/2018/08/facebooks-app-review-and-how-independent-research-just-got-a-lot-harder/

Sandvig, C. (2013). The internet as infrastructure. In W. H. Dutton (Ed.), *The Oxford handbook of internet studies (pp. 86–108)*. Oxford University Press.

Sayer, L. C. (2005). Gender, time and inequality: Trends in women's and men's paid work, unpaid work and free time. *Social Forces*, *84*(1), 285–303. https://doi.org/10.1353/sof.2005.0126

Schneider, N., & Mannan, M. (2021). Exit to community: Strategies for multi-stakeholder ownership in the platform economy. *Georgetown Law Technology Review*, *5*(1), 1–71. https://doi.org/10.31219/osf.io/nmyxp

Scholz, T., & Schneider, N. (Eds.) (2016). *Ours to hack and to own: The rise of platform cooperativism, a new vision for the future of work and a fairer internet*. OR BOOKS.

Sengers, P., Boehner, K., David, S., & Jofish, K. J. (2005). Reflective design. In: *Proceedings of the 4th decennial conference on critical computing: Between sense and sensibility* (pp. 49–58). Association for Computing Machinery. https://doi.org/10.1145/1094562.1094569

Shilton, K., Koepfler, J. A., & Fleischmann, K. R. (2013). Charting sociotechnical dimensions of values for design research. *The Information Society*, *29*(5), 259–271.

Star, S. L. (1999). The ethnography of infrastructure. *American Behavioral Scientist*, *43*(3), 377–391.

Star, S. L. (2010). This is not a boundary object: Reflections on the origin of a concept. *Science, Technology & Human Values*, *35*(5), 601–617. https://doi.org/10.1177/0162243910377624

Stoll, C., Klaaßen, L., & Gallersdörfer, U. (2019). The carbon footprint of bitcoin. *Joule*, *3*(7), 1647–1661. https://doi.org/10.1016/j.joule.2019.05.012

Strzalkowski, T., Harrison, T., Sa, N., Katsios, G., & Khoja, E. (2019). GitHub as a social network. In T. Z. Ahram (Ed.), *Advances in artificial intelligence, software and systems engineering* (Vol. *787*, pp. 379–390). Springer. https://doi.org/10.1007/978-3-319-94229-2_37

Suchman, L. (2002). Located accountabilities in technology production. *Scandanavian Journal of Information Systems*, *14*, 16.

Werdmüller, B. (2018, April 26). I'm done with syndication: Let's help people be themselves on the Web. *Ben Werdmüller*. https://werd.io/2018/im-done-with-syndication-lets-help-people-be-themselves-on

Whittle, J. (2019). Is your software valueless? *IEEE Software*, *36*(3), 112–115. https://doi.org/10.1109/MS.2019.2897397

Yu, E., & Koh Ping, C. (2021, May 24). China's latest crackdown on bitcoin, other cryptocurrencies shakes market. *Wall Street Journal*. https://www.wsj.com/articles/chinas-latest-crackdown-on-bitcoin-other-cryptocurrencies-shakes-market-11621853002

Zannettou, S., Bradlyn, B., De Cristofaro, E., Kwak, H., Sirivianos, M., Stringini, G., & Blackburn, J. (2018). What is gab: A bastion of free speech or an Alt-Right echo chamber. In: *Companion proceedings of the Web conference, 2018* (pp. 1007–1014). International World Wide Web Conferences Steering Committee. https://doi.org/10.1145/3184558.3191531

Zuckerberg, M. (2018). *Hearing before the United States House of Representatives Committee on Energy and Commerce*. House of Representatives Committee on Energy and Commerce.

Author biographies

Jack Jamieson (PhD, University of Toronto) is a Postdoctoral Researcher at NTT Communication Science Laboratories. His research interests include inclusive design, values in design and software development, social media, and digital labor.

Naomi Yamashita (PhD, Kyoto University) is a Primary Researcher at NTT Communication Science Laboratories and a Visiting Professor at Kyoto University. Her research interests include computer-supported cooperative work and computer mediated communication.

Rhonda McEwen (PhD, University of Toronto) is Vice-Principal, Academic and Dean, at the University of Toronto Mississauga, and Canada Research Chair in Tactile Interfaces, Communication and Cognition. Her research interests include human-machine communication, cognitive effects of media, tactile interfaces, tablet communication, cognitive informatics, device mediation, and virtual reality.