

Social.Wiki: A Web Held in Common

Theia Henderson
MIT CSAIL
Cambridge, MA, USA
theia@mit.edu

Carmel Schare
MIT CSAIL
Cambridge, MA, USA
schare@mit.edu

Ana Dodik
MIT CSAIL
Cambridge, MA, USA
anadodik@mit.edu

Clemens N. Klokmoose
Aarhus University
Aarhus, Denmark
clemens@cs.au.dk

Ziv Epstein
MIT College of Computing
Cambridge, MA, USA
ziv@mit.edu

David D. Clark
MIT CSAIL
Cambridge, MA, USA
ddc@csail.mit.edu

David R. Karger
MIT CSAIL
Cambridge, MA, USA
karger@mit.edu

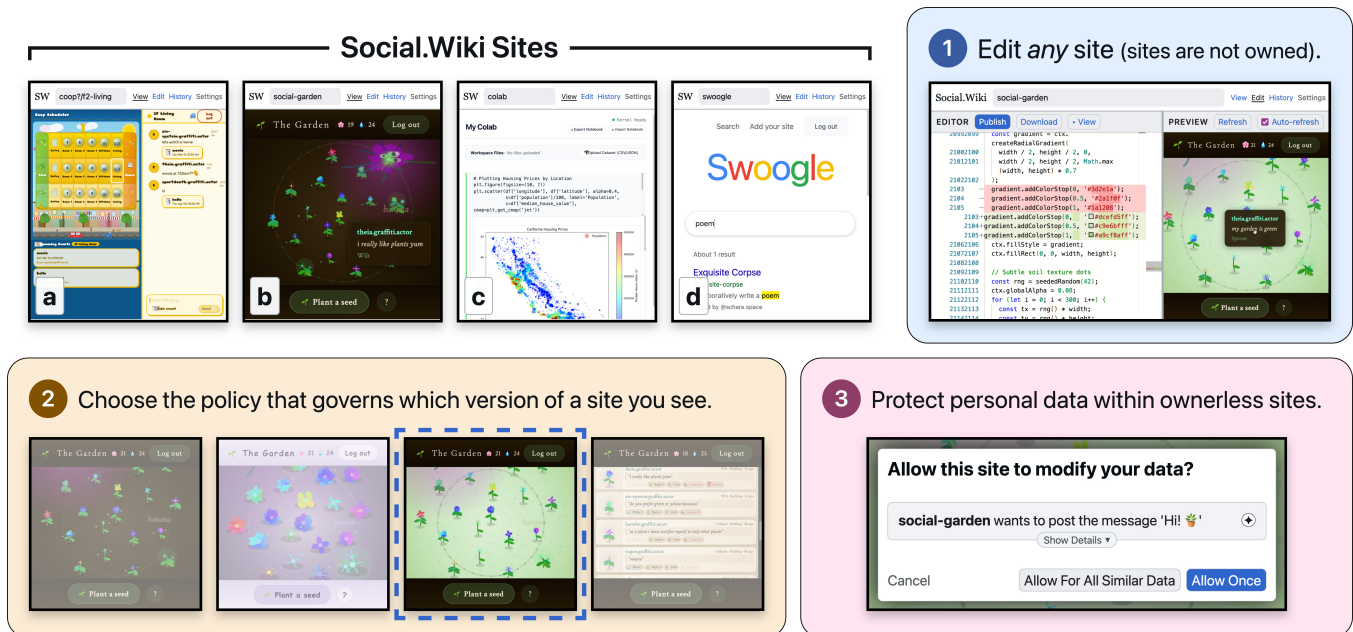


Figure 1: An overview of Social.Wiki, a reimagining of the web where sites are collaboratively edited like Wikipedia articles. Examples of sites in the top left were built by participants in two hours at a Social.Wiki hackathon: (a) a messenger and event scheduler for a housing co-op; (b) a creative microblogging site where posts are visualized as flowers; (c) a collaborative Python notebook; (d) a socially constructed web directory.

Abstract

Many of the websites people depend on have owners whose interests are not fully aligned with their users. We address the root of this problem by presenting a reimagining of the web where sites are *not owned at all* but are instead collaboratively produced like

Wikipedia articles. We call the system Social.Wiki because it supports the co-creation of interactive *social* sites, such as those for microblogging, messaging, dating, gaming, ride sharing, and so on. With off-the-shelf AI tools, people with little or no programming experience can edit these sites to better reflect the needs and preferences of their communities.

Social.Wiki builds on ideas from collaborative malleable software systems such as Webstrates, but is designed for public participation rather than use only within small, trusted groups. To this end, Social.Wiki includes governance to mitigate conflict. To accommodate diverse governance preferences, our model of “*plural governance*”



This work is licensed under a Creative Commons Attribution 4.0 International License. <https://creativecommons.org/licenses/by/4.0/>

© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2856-3/26/11
<https://doi.org/10.1145/3830398.3830639>

lets people independently choose the policies that determine which edits to a site they see. Social.Wiki also implements a granular security model to protect personal data in a malleable environment.

Complementing the decentralized design and governance of Social.Wiki sites, both site edits and within-site data are stored on Graffiti, a decentralized infrastructure, decoupling the ownership of underlying servers from the ownership of sites.

We evaluate Social.Wiki through case studies that demonstrate the range of sociotechnical structures it supports, as well as through deployments at a hackathon and in the wild.

CCS Concepts

• **Human-centered computing** → **Collaborative and social computing systems and tools**; **Collaborative content creation**.

Keywords

malleable software, peer production, plural governance, decentralization, content moderation, platform decay, end-user programming

ACM Reference Format:

Theia Henderson, Carmel Schare, Ana Dodik, Clemens N. Klokrose, Ziv Epstein, David D. Clark, and David R. Karger. 2026. Social.Wiki: A Web Held in Common. In *The 39th Annual ACM Symposium on User Interface Software and Technology (UIST '26)*, November 02–05, 2026, Detroit, MI, USA. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3830398.3830639>

1 Introduction

Sites on the World Wide Web have *owners*. At first, this design decision might seem inevitable, both as a technical means of delegating the “server” of a particular site and as a social means of governing the web. However, it is arguably because of this ownership model that most people are mere tenants within the sites they must inhabit to live their lives—subject to impersonal and often hostile designs simply to coordinate carpools, advertise their businesses, or connect with friends.

This work offers a reimagining of the web where all sites are instead *held in common*. On this web, any visitor to a site may also participate in the design of that site itself. This not only gives people the freedom to remove harmful “dark patterns,” but also allows communities to evolve the sites they rely on into spaces that are highly personalized to their needs, preferences, and values.

Our prototype of this web-in-common is called Social.Wiki. Its client interface, shown in Figure 1, combines elements of both a web browser and Wikipedia. Like the web, one can browse Social.Wiki to find all sorts of sites, including, importantly, *social* sites such as those for messaging, microblogging, meeting neighbors, dating, playing games, and so on. Like Wikipedia, every Social.Wiki site has an “Edit” button that any visitor can use to modify that site’s content, styling, and functionality.

If Social.Wiki were implemented like a traditional wiki, with our central server storing the edits, then it would be making the problem worse: exchanging the owners of individual sites with ourselves as the owners of *all* sites. Instead, Social.Wiki is implemented on top of Graffiti [33], a decentralized social infrastructure. Effectively, each edit to a Social.Wiki site produces a new version, stored on a server of the *editor’s choice*. When someone visits a Social.Wiki site, their client retrieves information about these versions from

the various chosen servers and selects one version to download and present. Thus, just as no one owns the web in its entirety, no one owns Social.Wiki. Admittedly, to provide a convenient entry point into Social.Wiki, our reference Social.Wiki browser is published on the (world wide) web at a domain that we do own¹. However, this browser is open source² and other competing browsers can be built and distributed, just like alternative web browsers.

A system where anyone can edit any site immediately raises the potential of destructive vandalism and conflict. The typical response in “peer production” [7] systems is to introduce a policy that governs changes to the editable artifacts. However, this work is motivated by the critique of one such policy—ownership. So, rather than replacing one normative means of governing an entire web with another, Social.Wiki draws on pluralist views of design and governance [4, 14, 20, 64, 80] to introduce what we call “*plural governance*,” which allows for multiple, possibly conflicting governance policies to coexist. Different coexisting governance policies may select different versions of a site to present.

Whichever governance policies people choose, no policy can catch every harmful edit to a site. Therefore, we also present security safeguards that protect against the damage that a malicious (or simply buggy) site version could otherwise inflict by destroying, leaking, or forging personal data.

By providing governance and security, Social.Wiki extends “computational media” to public participation settings. Computational media is a model of collaborative, malleable software where the boundary between user and developer is blurred [9]. Prior systems, such as Webstrates [43], rely on self-proclaimed “toilet door security” that does not scale beyond small, trusted groups [9].

1.1 Governance Overview

Like the web, each Social.Wiki site may contain *its own* governance for within-site interactions: a microblogging site may have a mechanism for flagging posts, and a messaging site may have a mechanism for blocking message requests. The governance that Social.Wiki provides is a layer above: it regulates the editing of *sites themselves*—including the editing of a site’s internal governance.

By default, the Social.Wiki browser provides a governance policy inspired by Wikipedia’s battle-tested system, but plural governance lets people freely replace it for themselves. Governance can be replaced browser-wide, or sites can act as localized governance “lenses” that provide views into alternative governance systems. For example, the site “Mypage,” described in Section 6.2, reproduces the governance of ownership by filtering a site’s edits for those by a single person (its “owner”). Mypage provides a space for people to create and view personal homepages but without imposing ownership on the system as a whole.

1.2 Personal Data Ownership

This work argues for eliminating the ownership of *shared social spaces*. Ownership of such spaces gives owners power over participants, including the power to determine who may participate and under what conditions. However, we actively work to protect the

¹<https://social.wiki>

²<https://github.com/sportdeath/socialwiki>

ownership of personal data because this form of ownership establishes an opposite social relation: rather than granting power over others, it protects an individual’s agency within a shared space.

In the traditional web model, the responsibility of protecting personal data is delegated to the owners of the sites that the data is produced on. The outcome is often disappointing, and in Social.Wiki, there are no owners to whom we could delegate such responsibility, even if we wanted to. Instead, all of a user’s data is stored on Graffiti (again, on the server of their choice) and all changes to that data are mediated by a client-side “*data guard*.” Just as a web browser mediates access to a camera or microphone, the guard confirms new data operations with the user before executing them—“Allow this site to modify your data?” This mediation prevents Social.Wiki’s openly editable sites from silently accessing, creating, or deleting personal data, whether through malicious intent or, more innocently, due to bugs from inexperience or “vibe coding.”

2 Motivation

This work is motivated by problems arising from the private ownership of a certain class of websites that represent shared social spaces, often called *platforms* [83]. Although Social.Wiki is a general system for collaboratively producing *many* kinds of sites, it notably supports the co-production of platforms. As such, it demonstrates a new model for how platforms might be created and governed without private ownership. This section reviews the problems associated with privately owned platforms and situates Social.Wiki within the broader landscape of proposed solutions.

Per Srnicek [83], we consider platforms to be infrastructures that facilitate digital connections between people. These platforms can be used for socializing, entertainment, conducting transactions, and organizing workplaces, to name a few. Examples include Facebook, Discord, Tinder, Reddit, Uber, Etsy, and Spotify. Owning a platform implies the “ownership of software (the 2 billion lines of code for Google, or the 20 million lines of code for Facebook) and hardware (servers, data centres, smartphones, etc.)” [83], as well as the data extracted from users [16]. The default ownership structure of today’s web is private ownership, which puts the power to design platforms into the hands of private individuals or shareholders. Notably, this leaves most users out of the design process.

The power asymmetry between platform owners and users can lead to harms. In some cases, owner decisions raise macro-scale concerns about user privacy [16, 56], the commodification of sociality [8, 15, 37], or disproportionate harms to minoritized communities [29, 53]. Other times, decisions can manifest as deceptive “dark patterns” [25, 54] or labor exploitation [46, 60, 61]. Recent years have seen an increase in these trends due to network effects [74, 83] that entrench dominant platforms, a dynamic described as platform decay [3] or more popularly “enshittification” [18, 19].

Many proposed solutions to platform harms involve the creation of structures that encourage platform owners to be more accountable to their users. Sometimes, these structures are self-imposed, by incorporating the platform as a nonprofit [90], or building it upon an open protocol such as ActivityPub [45], the AT Protocol [42], or Graffiti [33], each of which makes it easier to leave one platform for another. Alternatively, these structures can be externally imposed via legislation, such as the Digital Markets Act [21].

While owner accountability measures may mitigate the most egregious harms, and there are cases of long-term success (e.g., Vermont’s Front Porch Forum, Archive of Our Own [71]), they leave the underlying power distribution intact. The owners still retain the ultimate control over the design and governance of the platform, while users must rely on those owners to remain responsive despite changing leadership, incentives, or market pressures that can drive platform decay [65]. Users retain the ability to leave—made easier by open protocols that let them keep their data [33, 42, 45]—and can influence the platform by threatening to do so [35]. However, leaving still means abandoning the existing system rather than improving it. In contrast, we are interested in enabling users to *collectively* work to construct better platforms.

Platform cooperativism is an alternative model where a platform is collectively owned by its participants [65, 81]. However, platform cooperatives are difficult to produce because they bootstrap on the legal system; the first author knows from firsthand experience how difficult it is to create a legal cooperative in the United States.

Instead of having to rely on a patchwork of legal frameworks, Social.Wiki can be used to create new platforms that are collectively designed and governed by *default*. Ultimately, any Social.Wiki site *could* be constructed by building on an open protocol and crafting a bespoke legal solution with commons-based licenses and co-ownership guarantees. Our system unifies these demands into one system that supports the creation of cooperative platforms through *technical* means (decentralized hosting, production, and governance) rather than through legal means.

In some ways, these technical changes actually *simplify* the process of making a website by removing the need to purchase and configure a domain name or deploy a server. This ease was noted by participants in our hackathon (\$5.1). This makes it possible to build experimental new platforms “cooperative-first” rather than cooperativism being a reaction to already established power.

Some platforms cannot exist efficiently in Social.Wiki. In particular, aside from the ability to query a collective database, all computation happens on the *client*. This is fine for platforms that are scoped in some way (follow-based microblogging, location-based services, group chats) but not for algorithmic feeds or recommendation systems that must consume vast amounts of data. This limitation is inherited from the Graffiti system on which we build, and future workarounds are discussed by Henderson et al. [33].

3 Related Work

3.1 Peer Production

Peer production is a mode of creating goods and services through self-organizing communities rather than centralized control [6]. It is most prominently exemplified by free and open source software (FOSS) and Wikipedia, both of which demonstrate that loosely coordinated contributors can produce complex artifacts at scale.

This work builds off of the wiki model of peer production which provides a relatively simple means of direct editing. While FOSS projects include more expressive version management tools, like git, these tools can be complicated for novices [66].

Most peer production systems inevitably involve governance to maintain standards and resolve conflicts. While some are ruled by a “benevolent dictator for life,” [79], many adopt democratic practices.

However, no governance system is perfect (a point equally true in the physical world). After all, Wikipedia’s governance, while arguably the most successful, is still heavily debated 25 years on [2, 12, 32, 87]. For a system like Social.Wiki—which does not have the unified mission of creating an encyclopedia, but instead aims to house a myriad of different experiences and communities—we expect there to be even more disagreement.

Alternatively, remixing is a form of peer production that avoids the problem of governance because editors create independent derivatives rather than contribute to shared artifacts. To our knowledge, all systems for direct editing social apps use a remix system, including Scratch [73] and Websim [86], avoiding the question of choosing one system to govern a diversity of social apps. However, remix has the drawback of diffusing collaboration between non-canonical artifacts: on Scratch, only 3% of remixes build upon another remix rather than building on a fresh project [34], whereas on Wikipedia, the median article passes through 14 rounds of editing by different contributors, and highly visible articles undergo hundreds of rounds of editing [41].

With plural governance, this work attempts to strike a middle ground that does not diffuse collaborative energy but at the same time does not impose a fixed governance model on all participants.

3.2 Malleable Software

Malleable software is software that can be reshaped with minimal friction to suit individual needs [47]. Canonical historical examples are Smalltalk [40] and HyperCard [23], but spreadsheets are arguably the most durable example of malleable software currently used by the general public.

Editing malleable software has always been a collaborative practice, but in the early days this was achieved by trading floppy disks. More recently, malleable software systems offer the ability to collaboratively edit software in realtime over the internet. This trait was introduced by Croquet [82], a Smalltalk-based collaborative virtual 3D world, then popularized by Google Sheets, and generalized by Webstrates [43], a system that persists and shares changes to web pages in realtime. As the web platform is quite powerful, Webstrates makes it possible to collaborate on software for document editing, games, video conferencing, and so on [10, 27, 43, 70]. A decentralized implementation, called MyWebstrates, lets people collaborate without a centralized server [44], and the related system Patchwork introduces git-like versioning [48].

Realtime collaboration in a computational environment blurs the distinction between data and code. For example, in a Webstrates chat app there is no difference between sending a message and adding the message to the app’s source code. While this model is elegant, it obscures the semantics of what the change is for: is it a routine message, or is it a deeper change to the app’s functionality? Notably, this makes it impossible to release a Webstrate on the open internet without either making it read-only—disabling both malleability *and* social interactivity—or making it publicly writable and therefore vulnerable to abuse.

Social.Wiki extends the Webstrates vision of collaboratively editable websites by introducing governance and security properties that aim to make it feasible to widely publicize sites while maintaining malleability and social interactivity. Doing so involves a

fundamental change to the architecture that (re)introduces a separation of code and data. Code is used to describe a site’s interface and functionality, whereas data, representing within-site user activity, is populated into the site during runtime via queries. Changes to the code can be *governed* without hindering regular user interactions like sending messages. Meanwhile, changes to data are made *secure* against possibly malicious sites that could otherwise delete important data or post under a user’s identity without permission.

Additionally, the separation of code and data extends Webstrates by making it possible for the data populated into a site to be specific to the person viewing it. For example, the messaging app we discuss in Section 6.1 displays only the conversations that the current user is part of, rather than *all* conversations created within the site—this is crucial for preventing disclosure of private conversations. Separating code and data also makes it possible to build apps that present slices of very large datasets, like the ride sharing app we describe in Section 6.3 or Social.Wiki itself, which is built recursively in its own malleable system.

The elephant in the room with malleable software is whether people have the skills, time, or courage to change software. This is arguably why spreadsheets maintain popularity as they have a much lower barrier to entry despite a lower ceiling of expressivity. However, AI is rapidly changing this picture. Many people who don’t consider themselves programmers now engage in “vibe coding” [67, 75], and a host of tools are being developed to streamline the process [11, 57]. With Social.Wiki, AI is not embedded in the system itself, but we have created a Social.Wiki site that provides a prompt that can be used with any AI chatbot or agentic coding tool to edit or generate Social.Wiki sites with little to no experience. This tool was used extensively in our Social.Wiki deployment (§5).

4 System Design

On the surface, Social.Wiki is a Wikipedia-like system for editing interactive websites rather than static articles. One can browse Social.Wiki either by entering the “name” of a site in the navigation bar or by clicking on a link to the site. Every site can be interacted with through three tabs: “View,” which simply presents the site, “Edit,” which allows one to modify the site as a document and publish new versions, and “History,” which allows one to navigate through different published versions of the site. The system is governed by a Wikipedia-like model where certain users called “trusted editors” can mark sites as protected, thereby limiting who has edit access.

Under the surface, the system is more abstract: View, Edit, and History are all *themselves* editable documents that we call “governance lenses.” Modifying the governance lenses changes how Social.Wiki sites are governed by changing which site versions are selected for display and how those versions are created and annotated. To avoid having to recursively govern the governance lenses, modifications to these lenses are only applied locally.

The root of the system is a browser. It provides Social.Wiki’s top-level navigation bar, a default set of governance lenses, and a simple interface for modifying the governance lenses (the Edit lens does not edit itself). The browser also establishes a connection to a database. The governance lenses use this database to publish and fetch site versions while sites themselves might use the database to publish and fetch social data like posts, messages, and reactions. We

use the Graffiti system as the database because it is decentralized and provides a minimal-but-expressive client-side API that makes it possible to build a wide range of apps [33].

Different versions of the same site “interoperate” because they draw their social data from the same underlying database. Therefore, competing governance lenses can present different site versions without isolating their users from one another: posts on one site version can be read on another. The coexistence of different governance lenses is what we call “plural governance.”

The connection to the database is mediated by a client-side “data guard,” loaded by the browser, that surfaces access requests to the user so that users can securely browse collaboratively produced sites without those sites being able to destroy, leak, or forge user data. Additionally, every layer of “transclusion”—including a document within another document [43, 62]—is sandboxed so that sites cannot manipulate the governance lenses or the browser.

A diagram of the system is shown in Figure 2.

One notable feature of Social.Wiki is that the only active servers involved in its operation are the generic Graffiti servers. The rest of the system is a hierarchy of interactive *documents*. The browser is a document that uses Graffiti to dynamically fetch and transclude governance lens documents, which use Graffiti to dynamically fetch and transclude site documents. In other words:

Social.Wiki is a system of recursively transcluded interactive documents, where each document in the recursion is given guarded access to a shared database.

We classify the documents that fetch and transclude other documents as “lenses,” a concept borrowed from functional programming [22]. The governance lenses, as well as the browser, are lenses. Additionally, sites—which can be just as expressive as the browser or governance lenses—can themselves be used as lenses, providing localized governance views, as later demonstrated in Section 6.2.

The browser is currently distributed on the (world wide) web at <https://social.wiki> via the static hosting service GitHub Pages. However, it could just as easily be distributed over email or via a flash drive. In the future, others might develop alternative browsers, just like alternative web browsers. So long as these competing browsers connect to the same database and make similar queries, they will interoperate with the original. Therefore, at the ecosystem level, the browser is also modifiable, making the system plastic all the way through.

4.1 Graffiti: Serving Sites and User Data

Social.Wiki requires some sort of database to publish and query several types of content: the documents representing different proposed versions of a site; annotations of those documents to guide governance decisions, such as protection labels; and personal data displayed within sites themselves, such as messages and reactions. We choose to use a system called Graffiti as this database.

Graffiti provides a general-purpose client-side JavaScript API for publishing and fetching data with built-in design considerations specifically for *social* data such as attribution, access control, and context management. Additionally, the Graffiti API is implemented on top of a *decentralized* architecture—people can freely choose the provider that hosts their own Graffiti data or even self-host [33]. Importantly, no specific Graffiti server is used to host any specific

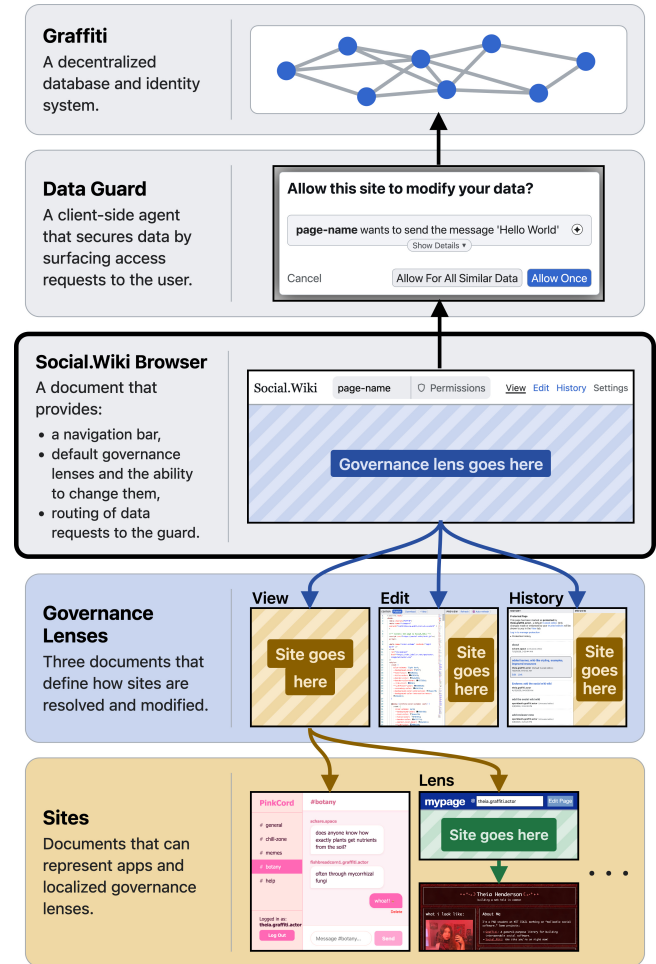


Figure 2: The Social.Wiki system. Using Social.Wiki starts by loading its *browser*. The browser connects to *Graffiti*, a decentralized database and identity system, via an intermediate *data guard* that confirms requests with the user. The browser also provides a navigation bar and three default *governance lenses*: View, Edit, and History. The View lens queries the Graffiti database to resolve the name entered in the navigation bar into a document that is displayed as a *site*. Sites may also query the database to populate themselves with posts, messages, and other user data. The browser provides a means of modifying the governance lenses, changing the governance of the system. Alternatively, sites can be used to resolve other sites, providing localized governance lenses.

Social.Wiki site. Instead, Social.Wiki sites are resolved by querying the *network* of generic Graffiti servers as a large virtual database.

Graffiti differs from other decentralized social infrastructures such as the AT Protocol [42] (underlying Bluesky) and Activity-Pub [45] (underlying Mastodon) because its client-side API is general enough that a wide variety of applications can be built on top of it with *no additional servers*. By comparison, building a custom application on the AT Protocol requires running a separate “App View”

server, and building a custom application on ActivityPub requires hosting a separate server “instance” [33]. By eliminating the need for additional servers, Graffiti allows a complete social application to be expressed within a single HTML document—the medium we choose for representing and editing sites on Social.Wiki³. This is especially important for Social.Wiki because it reframes “shared ownership of a social application” to be “shared ownership of an HTML document,” rather than “shared ownership of a server.” In doing so, we avoid the legal and physical complexities of jointly owning servers.

Graffiti data is managed *individually*: a person can post and delete their own data, but cannot either create data on behalf of someone else or delete someone else’s data. While this may seem counterproductive to our goal of common ownership, clients can flexibly interpret collections of individual data objects as shared interactions. For example, Social.Wiki sites are stored within Graffiti as a series of individually edited versions, but the View lens processes those edits to produce the experience of a collaboratively edited document. Henderson et al. [33] call this approach “total reification” as all interactions between people are reified into individual data objects. Importantly, total reification makes it possible to apply different moderation policies to the same underlying set of reified user actions. Social.Wiki’s plural governance is implemented via this property by applying different selection algorithms to the various proposed versions of a site.

Graffiti data is organized into “channels.” Like hashtags, channels are identified by arbitrary strings of text, do not need to be explicitly instantiated, and are *not owned*—anyone can post to a channel. A Social.Wiki site’s name is used as the channel for broadcasting edit metadata for that site. Importantly, it is not possible to find Graffiti data without explicitly querying one of its associated channels, a property designed to prevent “context collapse” [52], which in the case of Social.Wiki allows sites to be effectively “unlisted”: impossible to find unless given the exact name, especially if the name is long and random.

Graffiti provides Social.Wiki with substantial expressive power, allowing applications for messaging, microblogging, collaborative editing, and ridesharing to be represented entirely as HTML documents. However, Graffiti applications cannot ordinarily be edited by simply clicking a button. They are produced and governed just like other websites, arguably leading to the concentration of power outlined in Section 2. Social.Wiki instead enables people to openly collaborate on Graffiti applications and to govern that collaboration. Interestingly, Social.Wiki uses Graffiti to implement the lenses that facilitate collaboration and governance; however, this is not a straightforward feature of Graffiti but rather a novel use of Graffiti as a general-purpose database. Finally, we introduce a data guard that secures access to Graffiti. Although the data guard is particularly useful in Social.Wiki’s dynamic and ownerless environment, it is also applicable to Graffiti applications more broadly, as discussed in Section 4.3.2.

³While the Hypertext Markup Language (HTML) was originally developed to represent text documents, the addition of CSS for styling and JavaScript for interactivity makes HTML a rich graphical and computational environment. Webstrates [43] similarly uses HTML as its medium, using synchronization servers for data persistence and sharing (see §3.2), rather than Graffiti servers. HTML also has the benefit of running on practically any device due to the ubiquity of web browsers.

4.2 Plural Governance

The web is governed by a system of ownership. In removing that ownership, we are faced with the difficult decision of choosing an alternative system of governance for resolving conflicts and maintaining quality. While we do provide a specific *default*, based on Wikipedia’s governance, we propose the unusual approach of avoiding a concrete decision by allowing people to *choose their own governance*. This choice draws on pluralist approaches to design [4, 14, 20] and on governance scholarship that rejects the idea that any single normative governance model will ever be appropriate for all communities and situations [64, 80].

To put our plural governance in perspective, some amount of personalized governance is already essential to most social platforms via features such as “blocking,” which affects one’s own experience but not other people’s. However, block lists are laborious to maintain and therefore unsustainable as the only form of governance [38, 76]. Systems like SquadBox [51] and Trustnet [36] demonstrate how the labor of blocking people and posts can be reduced by delegating to close friends or other trusted sources, a feature now widely deployed in Bluesky under the name “composable moderation” [24], which allows one to subscribe to different moderation services. Social.Wiki generalizes the personalization of governance beyond layers of block lists to the choice between different governance *policies*, which may involve context-specific delegation, reputation systems, automation, and so on.

The crux of governing a web is the resolution of *names*. It is fairly trivial to create a system allowing anyone to publish (their versions of) documents under gibberish identifiers. What is challenging is deciding which of those documents should be displayed when one visits a site with a human-readable name, like “facebook”. The World Wide Web takes the relatively straightforward approach of hierarchically assigning these names to owners via the Domain Name System (DNS) [58] and having those owners choose how their names (URLs) are resolved⁴. In Social.Wiki, names are not owned. Instead, names are resolved independently by each user according to the policy described by their governance lenses.

To ground the governance possibilities in a concrete example, we first discuss how our default system of governance is implemented within the governance lenses. Then, we explore how those lenses can be modified to enable plural governance. Further consequences of plural governance are discussed in Section 7.

4.2.1 Default Governance. Our default governance assumes, as do many peer production systems, that most participants act in “good faith” [72], and therefore sites are openly editable by default (a “doocracy” [13, 85]). However, changes can be reverted via the History tab. In cases of prolonged disagreement or “edit wars” [84], sites can be marked as “protected” by a “trusted editor.” When a site is protected, only edits made by a trusted editor are displayed.

Even our default governance adds a degree of pluralism not found on Wikipedia: users may *choose* who their trusted editors are, so a site can be protected for some people and not for others. This property, inspired by the system Trustnet [36], is included because Social.Wiki is intended to contain many communities, and we do not expect any central body of administrators to know of every

⁴Decentralized alternatives to DNS like IPNS [5], Tor [17], and Namecoin [39] still assign names to owners but via cryptography rather than a central arbiter.

community’s existence, let alone have the context to moderate them. By allowing people to choose who they trust, the system has the potential to adapt to these small, disconnected communities.

The View lens implements this default by querying Graffiti for the logged-in user’s trust relationships plus two kinds of information based on the name provided in the browser’s navigation bar: published versions of the named site and any protection labels applied to it. Using trust and protection data, the lens determines whether the site is protected for that user. If so, it excludes versions published by people the user does not trust. It then selects the most recently published eligible version and transcludes its associated HTML document to render the site. If no eligible version exists, the View lens displays “Nothing here...yet” and links to the Edit lens.

Our default system of governance begs many questions: What if I trust someone to edit certain sites but not others? [69] Could the set of trusted editors be determined dynamically by a reputation system? [1] Could AI be used to automatically block edits based on their content? [31, 68] These can all be addressed by modifying the governance lenses.

4.2.2 Modifying Governance Lenses. The governance lenses are implemented as HTML documents, just like the sites they govern. The document used for each lens can be replaced within a Settings tab, provided by the Social.Wiki browser. Alternatively, lenses can be replaced by clicking on a link that encodes the new documents, making the process of changing the lenses a one-click experience like installing a browser extension. To mitigate errors, the browser warns the user before making a change and provides an option to reset to the default lenses.

As all of the underlying Graffiti data is “totally reified” (§4.1), alternative View lenses can reinterpret the site versions, protection labels, and trust relationships provided by our default lens. They can also query Graffiti for entirely new types of data—endorsements, flags, tags, *etc.*—which alternative Edit and History lenses can be modified to produce. One such alternative system might replace our default system of reactive site protection with *proactive* peer review where new edits to sites are not shown until endorsed by at least two trusted editors.

Changes to the governance lenses are only applied locally, providing a base case in what would otherwise be an infinite nesting of governance layers. However, there are cases where a user might want their governance updated on their behalf. This can be achieved by installing a governance lens that introduces another layer of indirection via an additional lens. Suppose, for example, that a user wants their Social.Wiki experience to follow rules chosen by their community’s leaders. Their View lens could look up the rules most recently endorsed by those leaders and transclude the HTML document that implements them as a lens. That lens would then determine which site to display, producing the nesting “governance lens → endorsed lens → site.”

We expect that most people will not develop their own governance lenses, instead choosing those created by others. As governance lenses are just documents, they can be shared via direct message, advertised in community spaces, or published in a hypothetical “governance store,” similar to an app store. Alternatively, a system like PolicyKit [89] could be adapted to create lenses from higher-level primitives. None of these distribution or lens-crafting

mechanisms require any new infrastructure to deploy—they can be built within Social.Wiki itself.

4.3 Security

If Social.Wiki’s governance were perfect, with all sites vetted by experts working in one’s interest, then we would already be at the end of our design story. Of course, as we have discussed, no governance will ever be perfect. While we hope that most of the time the governance that people choose for Social.Wiki is good enough not to expose them to written or visual content that they would consider harmful, Social.Wiki involves the co-production of *code* which introduces the possibility for small changes in that code to cause massive damage.

In particular, Social.Wiki requires dynamically executing other people’s code *and* giving that code access to read and write personal data. Both of these present possible security vulnerabilities, which we address here.

4.3.1 Sandboxed Transclusion. Dynamically executing untrusted code is a classic security risk because, without proper protections, the running code can observe and modify its execution environment. If this were the case within Social.Wiki, visiting a malicious site could inject code into the governance lenses or browser, which could in turn manipulate the governance, leak personal data, *etc.*

Fortunately, modern browser engines are already excellent at isolating execution environments. This isolation is what makes it possible to visit a website without that site modifying either the browser or sites running in other tabs. For our usage, browser engines also provide the ability to isolate transcluded documents: the `<iframe>` tag, which is used to include HTML within HTML, can be configured to run as a sandbox [55], completely isolating the transcluded code from its parent, except for a channel to pass messages between them.

Unfortunately, sandboxing leads to all sorts of restrictions, especially when sites do not have “origins”—Social.Wiki sites are fetched from Graffiti and so are transcluded by their HTML source rather than via an HTTP link. Most of our engineering efforts on this project involved wrestling with these restrictions. Functionality as simple as following links between sites has been reconstructed by bubbling navigation messages up through the `iframe` tree.

Every Social.Wiki document (browser, governance lens, or site) includes a line of code that injects the script `https://social.wiki/init.js`. This script establishes a message connection to the document’s parent and provides the web component `<sw-transclude>`, which can be used to transclude other documents. `<sw-transclude>` does the work of automatically configuring a sandboxed `<iframe>` and establishing a message connection to it.

4.3.2 Data Guard. Graffiti is used within Social.Wiki to publish sites, trust editors, send and receive private messages, and perform all other social actions within the system. Graffiti data is also interoperable, ensuring that social interactions persist across different versions and forks of a site and different governance systems. However, all of this expressivity and interconnectedness creates risk. Without proper protection, a malicious site could delete all of a person’s Graffiti data, publish forged messages on their behalf, leak private messages, and so on.

To mitigate this risk, sites are not given *direct* access to Graffiti at all. Instead, only the “data guard,” embedded in the browser, is given custody of the Graffiti credentials needed to modify data and read private data. To perform a Graffiti operation, a site sends a request to the browser, which in turn forwards the request to the guard, which confirms the request with the user via a pop-up, as shown in Figure 1, panel 3. The user is presented with an approachable summary of the request, generated by a local AI model [77], and can also inspect the exact request details. If the user allows the operation, the result is returned to the site.

Effectively, the data guard is guarding access to Graffiti, just like a web browser guards access to one’s camera or microphone. This granular security model, based on the Principle of Least Privilege [78], allows one to safely browse and even interact with random Social.Wiki sites, with the certainty that the sites will not modify or access any data other than the data they explicitly request and are granted access to.

Of course, for continued use of a site, these constant pop-ups will surely induce “confirmation fatigue.” To mitigate this, the pop-ups give the option to “Allow For All Similar Data.” This option grants the requesting site continued permissions to perform “similar” data operations until revoked or the site is changed⁵. Permissions can be revoked via the Permissions tab that appears in the browser’s navigation bar (visible in the browser in Figure 2).

Currently, data similarity is assessed via a deterministic set of rules that match data objects with the same property names and value types. However, in the future, a local AI agent could offer a more nuanced notion of similarity, perhaps even detecting messages that seem uncharacteristic for the sender.

For people developing Social.Wiki sites, the complexity of passing Graffiti requests to the data guard is abstracted away. The `init.js` script described in Section 4.3.1 exposes a global class called `window.Graffiti`. This class provides all the methods of a normal Graffiti implementation, only under the hood these methods actually send messages to the data guard.

The data guard is reusable in Graffiti apps outside of Social.Wiki. To achieve this, the guard’s code is hosted on its own domain and embedded into the target app, e.g., our browser, via an `iframe`. Anti-cross-origin-scripting features ensure that the embedding app cannot access the guard’s credentials. Effectively, the guard acts as an external server, even though its code runs entirely client-side.

5 Deployment

We gradually deployed Social.Wiki to friends and colleagues over two months and then hosted a publicly advertised hackathon. At the time of writing and based on the information available (it is a decentralized system) Social.Wiki has had at least 100 unique users.

To reduce the barrier to entry, the Social.Wiki landing page links to a guide that walks through “vibe coding” a Social.Wiki site. This involves copying our meta-prompt describing the system into an AI chatbot or coding agent. In a single shot, these systems can generate messaging apps, dating apps, board games, collaborative

canvases, and so on. While vibe coding can introduce security risks if used improperly, sites are sandboxed and requests to modify user data are surfaced to the user (§4.3). This makes it safe to interact with a site built by any amateur programmer, including a coding agent. Many, but not all, sites were at least partially vibe coded.

The sites that have received the most continued use so far are those that are in support of existing community groups, specifically a reading group and a film club. These sites both began as static sites with their schedules directly embedded in the HTML. However, over time—due to the efforts of multiple editors—each accrued social features including comments on each reading and a collectively curated watch list. Effectively, these sites evolved into “microplatforms” in direct response to user needs, and they can continue growing and evolving as they are used. There is no “webmaster” who is on the hook to implement desired changes or who has the privilege of dictating the final design.

Given that Social.Wiki usage is currently relatively small, we have not seen the full use of its governance affordances, which are instead evaluated through case studies (§6). However, we have observed two lighthearted disagreements. First, the film club site underwent a brief edit war when a user registered an anonymous Graffiti identity and changed the watch schedule to set *Twilight* as the next movie, triggering amusement in the film club’s group chat. Second, a chat app for passing notes in class, named simply “chat,” was forked to create a new app called “betterchat” that altered the interaction design so users could send multiple notes at a time.

5.1 Hackathon

Our hackathon was a three-hour event that attracted 12 participants. After a brief introduction to the system, people spent two hours building before presenting their work and filling out a survey.

At the end, three awards were given out: “Most Likely To Be Used,” to an internal housing co-op communication tool; “Most Creative,” to a microblogging site where posts are planted as flowers in a garden; and “Trust and Safety,” to an end-to-end encrypted Slack clone⁶. Other highlights include a collaborative Python editor⁷, a choose-your-own-adventure game where users can collaboratively add branches, a Social.Wiki search engine, a dating app for poly-cules, and a version of Conway’s Game of Life that is seeded by data posted to other sites. Some examples are shown in Figure 1.

11 out of 12 participants completed an exit survey. All self-reported some programming (median: 4/5, min: 2, max: 5) and front-end experience (median: 3/5, min: 2, max: 5). Future work could evaluate Social.Wiki for non-experts (although, due to AI, most participants were not “programming” in a traditional sense).

5.1.1 Developer Experience. A number of participants spoke positively of the development experience (P2, P4, P5, P7, P8, P9, P10), with P5 noting it was “Way easier and way faster than any other development experience I’ve ever had” and P8 noting “it felt powerful functional and value aligned!” However, there was also room for improvement: P2 and P10 wanted an improved editor, P4 and P1 wanted the ability to split development over multiple files, and P8 wanted better debugging. For the sake of time, participants were

⁵Note that the permissions system does not explicitly know that a site is changed because site changes are defined by the View lens. Instead, the permissions system tracks transcluded documents by IDs that are assigned recursively from parent to child documents. The View lens assigns IDs to its children based on the hash of their HTML documents, which makes a site’s ID change whenever the site’s content is changed.

⁶Note that Graffiti is not currently end-to-end encrypted, but there is ongoing work to do so [33], which would make all Social.Wiki sites end-to-end encrypted by default.

⁷Python can be executed in the browser using Pyodide.

not told about the extent of Social.Wiki’s plasticity, which in theory could be used to address these issues.

Participants appreciated how batteries-included the system was (P3, P7, P8, P10, P11); e.g., P8 liked “The rapid ability to prototype and not having to spend any time worrying about deployment, data models, servers”. P9 expressed a wish for more documentation and “Having trouble figuring out how to start my project.” However, P7 pointed out, “discoverability is simple and fast enough that someone can figure it out just by clicking around.” Finally, P5 and P9 expressed desires for a “directory that lets me see what’s out there and how pages relate to each other!” (P9). Fortunately, P11 built a social search engine (Fig. 1, d) that does exactly that!

5.1.2 Collaboration. Most hackathon participants decided to work alone and so did not take advantage of collaborative features. However, P11 noted, “It was cool to easily create a website that could be used by anyone and could be kept up even if not by me.” Despite editing alone, participants were observed regularly visiting and interacting with each other’s sites throughout the hackathon.

5.1.3 Permissions. Permissions were a noted pain point for P7, who compared them to “disallowing cookies on every site.” However, P5 said, “I love the feature of being able to see what you’re uploading (the “allow” button)” and wanted even more transparency on data being shared. This suggests that future work should provide means of delegating or automating more of these security decisions, similar to the cookie Consent-O-Matic system [63], while maintaining the granular control and transparency for others who want them.

5.1.4 AI. Participants were split over AI. Some thought that it made development easier (P2, P6, P7), and P5 liked using a chat-bot primed with our prompt as a tutor to learn about Graffiti and Social.Wiki. However, P1 found it frustrating and P6 felt it did not allow them to understand the “inner workings” of their application.

AI is not inherent to using Social.Wiki but was encouraged during the hackathon to ensure that people with a range of skills could build something interesting in a short amount of time. It is an open question as to what alternative low/no-code tools might be created for development and what role AI will play in the sites intended for long-term use rather than rapid prototypes.

6 Case Studies

6.1 OurChats: A Malleable Messaging Platform

OurChats is a messaging platform, akin to Facebook Messenger. What makes OurChats unusual is that each chat includes an “Edit” button that allows participants to modify that particular chat’s styling and interface features. For example, one OurChats chat has been modified to use rainbow Comic Sans font, have an animated snowy background, and include snow reactions, as shown in Figure 3a.

Under the hood, any time a new chat is created, it is actually creating a new *site* from a basic chat template. This site is transcluded within the OurChats site so that it can be displayed along with the OurChats sidebar that lists all chats.

OurChats demonstrates how the benefits of malleability can be integrated into an otherwise conventional chat interface—there is no need to manually create a new site from scratch just to form

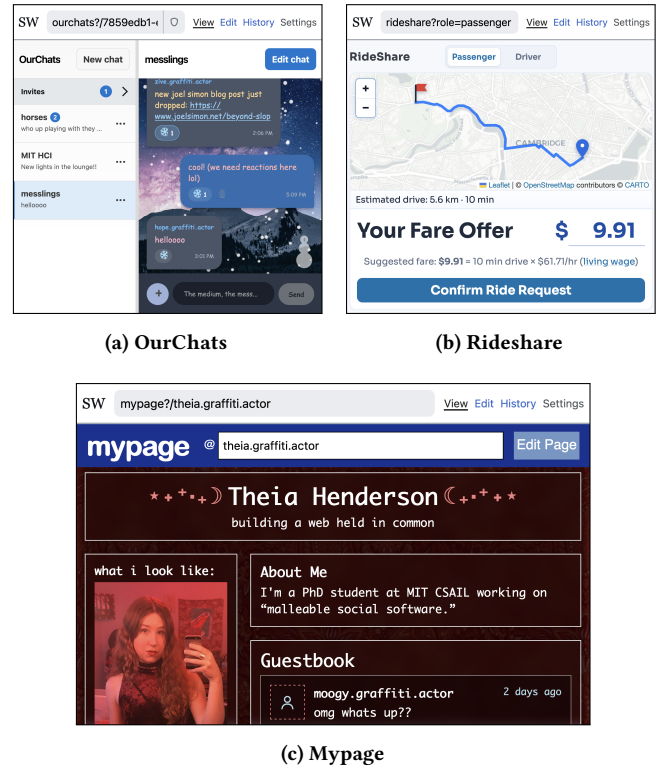


Figure 3: Screenshots from our case study sites.

a new conversation and no need to manually check a variety of independent chat sites for new messages. This per-chat malleability allows people to bring the features that they like to express themselves with [26] to any chat, without creating a complex super-app that has all the possible features of every chat.

Additionally, OurChats demonstrates how transclusion can be used to delineate governance boundaries. Regardless of whether OurChats is protected (or otherwise governed under other systems), the fact that the chats within OurChats are separate sites means that they are governed independently.

Subtly, transclusion within OurChats is used differently than transclusion in the View lens. The View lens runs a resolution algorithm (e.g., as in Section 4) that transcludes a document by *source*. OurChats does not reimplement that resolution logic. Instead, it transcludes by *reference*, bootstrapping off of the user’s chosen View lens rather than introducing new governance.

6.2 Mypage: A Page as a Governance Lens

Mypage is a site for making and browsing personal homepages, similar to Myspace or Neocities. You can visit someone’s “Mypage page” by entering their Graffiti username into a navigation bar. A Mypage page can include both static content and social features, like the guestbook in Fig. 3c. Importantly, unlike the rest of Social.Wiki, inside Mypage *you cannot edit someone else’s Mypage page*.

Under the hood, Mypage is a lens that works much like the View lens. It treats a username entered in the Mypage bar as a site name. Then, it uses Graffiti to look up edits associated with that name and

transcludes the document corresponding to the most recent edit. However, unlike the View lens, Mypage only considers edits to a site by *one person*—the person whose username was entered.

Mypage produces a governance of *ownership*. While we began this paper critiquing the ownership of *shared social spaces*, there is of course still value in having ownership over one’s personal identity, as clarified in Section 1.2. Mypage allows one to express their personal identity through an interactive webpage. However, if Mypage were used to create a community platform and there was a falling-out between community members and the creator, members would not be forced to experience the creator’s “personal expression.” Instead, they could remove the Mypage lens and view the page within Social.Wiki as a whole, where it is openly editable.

Mypage also demonstrates a routing convention respected by the default View lens and other lenses. In the name ‘mypage?/user,’ the text ‘/user,’ following the special character ‘?’, is not treated as part of the site name but instead is sent as a parameter to the transcluded site. This makes it possible to directly link to a particular page within a site, similar to client-side routing in single-page apps.

Finally, note that even though Mypage does not allow people to edit each other’s Mypage pages, anyone can edit Mypage itself. Therefore, if Mypage were to become popular, it might need to be governed (via protection or otherwise) to prevent people from “breaking” Mypage’s governance, which could (among many other possibilities) be used to vandalize Mypage pages. Of course, “breaking” is subjective here as there is no reason why the name Mypage *must* be associated with this particular functionality.

6.3 A Collectively Owned Ride-Sharing App

“Rideshare” is a Social.Wiki site for matching riders to drivers, much like Uber or Lyft (see Fig. 3b). A rider selects where they want to be driven from and to. Then, the rider is presented with a suggested fare based on a transparent formula that multiplies the expected time it will take to drive along the route by a local “living wage,” per the Living Wage Calculator [49]. The rider can adjust the fare they want to pay for the ride before submitting a request.

In a separate pane, drivers can browse for nearby ride requests. The exact pickup and drop-off locations of the requests are not revealed to the drivers for privacy purposes, but drivers can see approximate locations as well as the fares being offered. Once a driver accepts a given request, the rider is asked to confirm the driver, at which point the exact pickup and drop-off locations are revealed to the driver and the trip can commence. Like a traditional taxi service, fare is expected to be paid directly by the rider to the driver at the end of the ride, via cash or digital payment.

Rideshare uses the peer-produced map service OpenStreetMap [30] for displaying interactive maps and calculating routes between locations. To efficiently query ride requests in a given location, ride requests are indexed spatially into Graffiti channels according to an open-source global grid [28] (interestingly, developed by Uber).

Rideshare demonstrates how Social.Wiki can be used to produce a multi-stakeholder cooperative that is accountable to both the riders and drivers. While we can only speculate about how the competing interests will play out, it could result in more transparent matching algorithms, as well as adaptable policies for safety, dispute resolution, and working conditions. Additionally, Rideshare does

not extract any commission because it has no owners. Of course, Rideshare does not solve all of the problems of the gig economy. In particular, its decentralized nature means it cannot be used to offer a minimum wage or health insurance. However, arguably these services might be better provided directly by public-benefit organizations like governments, unions, or mutual aid groups, rather than by employers who must be coerced into doing so.

So far, we have used Rideshare to successfully complete one ride. The ride was performed voluntarily rather than for pay because it is not clear how Rideshare fits into existing local regulation: the companies that *own* ride sharing platforms must be licensed, but what does that mean for a platform that is not owned?

For realistic use, Rideshare will of course need to be upgraded to include safety features such as driver and passenger reputation services and the ability to opt into different background check services, both of which are more than possible within Social.Wiki.

One technical note is that the sandboxing introduced in Section 4.3.1 does not allow Social.Wiki sites to access one’s location—in Rideshare, one’s starting location is provided manually. In the future, the data guard used to grant access to Graffiti could be expanded to match the browser mechanism that inspired it by guarding other inputs like one’s location, camera, or microphone.

7 Discussion: A Plural Web

Plural governance makes it possible for different people to see different versions of the same site. From an individual perspective, this empowers people to shape their experiences to their own needs and preferences. At the ecosystem level, however, this plural governance raises several natural concerns: it might cause confusion, diffuse collaborative effort, or erode the effectiveness of governance as a whole. In this section, we argue that the divergence that plural governance makes possible will be minimal, as Social.Wiki provides many intermediate opportunities to accommodate disagreement. Furthermore, we argue that, even when divergence occurs, confusion and diffusion can be mitigated and governance is not made ineffective.

As a caveat, these arguments have not been evaluated at scale. As we have deployed Social.Wiki to only roughly a hundred participants (see Section 5), we have fortunately not seen cases where even our default page-protection system was necessary, let alone cases where governance diverged. Scaling Social.Wiki to the point where these mechanisms become relevant must be done cautiously, by cultivating cultures of good-faith collaboration within communities [59, 72] and refining our reference browser’s default governance to provide an effective baseline.

7.1 Accommodating Difference

Governance in general is a means of last resort when disagreement cannot otherwise be accommodated. As Social.Wiki sites are interactive, interoperable software, they can accommodate more difference than if they were simple static articles. This creates a spectrum of intermediate forms of divergence before conflicting governance policies sever people’s views completely.

One approach is to turn a point of disagreement into a within-site *setting*. For example, a site does not strictly need to be in either light mode or dark mode: it can allow each person to toggle between

the two. A user’s personal settings may be stored, like any other data, on Graffiti. If there is controversy over the default settings, the site can instead present newcomers with a setup wizard containing multiple options.

If attempts at compromise fail, despite settings or traditional good-faith collaboration strategies [72, 88], one option short of divergent governance is to “fork” a site, *i.e.*, copy its source code under a new name. Since social data is stored on Graffiti rather than within a site, competing forks can continue to interoperate with the same data even as their code diverges. Unfortunately, a fork no longer automatically benefits from contributions to the original, potentially diffusing collaborative energy, similar to platform exit as discussed in Section 2 and remix systems as discussed in Section 3.1. This diffusion can be reduced by decomposing a site into common components shared between competing versions through transclusion, a form of modularity often used in Webstrates [43].

If there is disagreement over who should keep the name of a forked site, one amicable solution is for *no one* to keep it and instead replace the site at the original name with a disambiguation page linking to the competing versions. This is similar to how Wikipedia uses disambiguation pages for words with multiple meanings. However, the flexibility of Social.Wiki means that a disambiguation page need not be a simple list of links; it could instead be interactive, like a BuzzFeed-style quiz: “Which site is right for you?”

7.2 When Governance Diverges

If disagreement is so strong that people cannot even agree on a disambiguation page, governance must finally be used to choose a side. Different governance systems may make different choices, causing the same name to resolve to different site versions for different people. This divergence may make it confusing to discuss a site, since its name no longer refers to the same content for everyone. In such cases, people can link directly to a particular site *version*—each version automatically receives a long, hash-based identifier—rather than relying on a contested human-readable name. Alternatively, one’s View lens could surface ambiguities with a warning such as “Other people see different content than you” and a toggle for switching to a competing version. People can also share their governance lenses as sites, just like Mypage (§6.2), allowing others to temporarily view Social.Wiki through their eyes.

We conjecture that differences among governance systems will often resemble those between the “nightly” and “stable” releases of open-source software. Some systems may receive updates almost immediately but occasionally encounter broken or problematic content, while others may update more slowly after changes have been vetted through various processes. Systems that otherwise present completely disconnected realities give up the benefits of collaboration, creating pressure to converge where possible.

However, even if governance policies diverge completely, from an *individual’s* perspective, their chosen governance policy continues to work exactly as intended, protecting them from harmful or unwanted content even if it exists elsewhere. Social.Wiki can reproduce both Wikipedia-like protection and traditional web-style ownership; pluralism does not make either of these highly scalable models any less effective for those who choose them.

7.3 Governing Reach, Not Existence

Aside from external laws that make it illegal to distribute certain types of content, Social.Wiki’s decentralized design, governance, and infrastructure make it largely impossible to prevent harmful content from existing somewhere in the system. While this may seem alarming, it is no different from the existing web where it is almost always possible to host such content on an obscure domain.

This does not mean that Social.Wiki is neutral toward harmful content. Rather than pretending that such content can be made impossible to publish, Social.Wiki democratizes the design of content *reach*: communities can determine how and what they endorse, flag, and help others discover. We believe that distributing this power can foster healthier ecosystems of community care that discourage harmful content from being produced and spread in the first place. We cannot guarantee that the results will always be fair or effective, as no system can, but instead give the people affected by governance a role in designing and revising it, rather than placing that authority in the hands of an arbitrary external owner.

8 Conclusion and Future Work

The ownership of websites made sense in the web’s early days when it was largely used to browse personal homepages. However, as web traffic increasingly concentrates on a handful of dominant platforms, the owners of those platforms are “becoming owners of the infrastructures of society” [83]. By combining ideas from wiki-style peer production and collaborative malleable software systems, Social.Wiki imagines an alternative web where people can collectively reshape both the platforms they inhabit and the policies governing that collaboration.

While Social.Wiki is a functional and deployed system, our base of a hundred or so users is tiny compared to the existing web. Therefore, we have yet to see whether its system design or model of plural governance works at scale. Future work is also needed to refine our default governance policy (many people never change defaults [50]); relax Social.Wiki’s exclusively client-side architecture so that it can support computation-intensive tasks like algorithmic “For You” feeds or sites with millions of edits; improve the developer experience beyond single-file HTML editing for large, long-lived projects; and foster a culture of good-faith collaboration.

Social.Wiki does not eliminate conflict or the need for governance. Instead, it changes who has the power to respond to these challenges. Rather than leaving people as tenants within software designed and governed by distant owners, Social.Wiki gives communities a substrate for constructing, contesting, and continually revising their shared online spaces.

Acknowledgments

We would like to thank all of the hackathon attendees, with a special mention to Barish Namazov, Andre Ye, and Ryan Yen, whose creations are displayed in Figure 1. Additionally, the first author would like to thank Catherine D’Ignazio for her class on “Cities Without Property,” which inspired this work; Benjamin Mako Hill for insights on peer production; and Helena Vasconcelos for figure guidance. A. Dodik acknowledges the generous support from Schmidt Sciences and from National Science Foundation grant IIS2335492.

References

- [1] B. Thomas Adler and Luca de Alfaro. 2007. A content-driven reputation system for Wikipedia. In *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*. ACM, 261–270. doi:10.1145/1242572.1242608
- [2] Leah Ajmani, Nicholas Vincent, and Stevie Chancellor. 2023. Peer Produced Friction: How Page Protection on Wikipedia Affects Editor Engagement and Concentration. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW2 (9 2023), 1–33. doi:10.1145/3610198
- [3] Michael J. Ardoline and Edward Lenzo. 2025. The cognitive and moral harms of platform decay. *Ethics and Information Technology* 27 (7 2025), 37. doi:10.1007/s10676-025-09846-1
- [4] Shaowen Bardzell. 2010. Feminist HCI: Taking Stock and Outlining an Agenda for Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, 1301–1310. doi:10.1145/1753326.1753521
- [5] Juan Benet. 2014. IPFS: Content Addressed, Versioned, P2P File System. *arXiv preprint* (2014). doi:10.48550/arXiv.1407.3561
- [6] Yochai Benkler. 2002. Coase's Penguin, or, Linux and "The Nature of the Firm". *The Yale Law Journal* 112, 3 (2002), 369–446. doi:10.2307/1562247
- [7] Yochai Benkler and Helen Nissenbaum. 2006. Commons-based Peer Production and Virtue. *Journal of Political Philosophy* 14, 4 (12 2006), 394–419. doi:10.1111/j.1467-9760.2006.00235.x
- [8] Grant Bollmer and Katherine Guinness. 2024. *The Influencer Factory: A Marxist Theory of Corporate Personhood on YouTube*. Stanford University Press. doi:10.1515/9781503638808
- [9] Marcel Borowski, Bjarke V. Fog, Carla F. Griggio, James R. Eagan, and Clemens N. Klokmoose. 2023. Between Principle and Pragmatism: Reflections on Prototyping Computational Media with Webstrates. *ACM Transactions on Computer-Human Interaction* 30, 4 (8 2023), 1–53. doi:10.1145/3569895
- [10] Marcel Borowski, Jens Emil Sloth Grønbaek, Peter W. S. Butcher, Panagiotis D. Ritsos, Clemens Nylandsted Klokmoose, and Niklas Elmquist. 2025. Spatialstrates: Cross-Reality Collaboration through Spatial Hypermedia. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. ACM, 1–14. doi:10.1145/3746059.3747708
- [11] Yining Cao, Peiling Jiang, and Haijun Xia. 2025. Generative and Malleable User Interfaces with Generative and Evolving Task-Driven Data Model. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. ACM, 1–20. doi:10.1145/3706598.3713285
- [12] Viktorija Car and Dijana Šobota. 2025. Disinformation as a tool for digital political activism: Croatian Wikipedia and the case for critical information literacy. *Journal of Documentation* 81, 5–6 (8 2025), 1145–1162. doi:10.1108/jd-01-2025-0020
- [13] Katherine K. Chen. 2009. *Enabling Creative Chaos: The Organization Behind the Burning Man Event*. University of Chicago Press. doi:10.7208/chicago/9780226102399.001.0001
- [14] Sasha Costanza-Chock. 2020. *Design Justice: Community-Led Practices to Build the Worlds We Need*. The MIT Press. doi:10.7551/mitpress/12255.001.0001
- [15] Nick Couldry and Ulises A. Mejias. 2019. *The Costs of Connection: How Data Are Colonizing Human Life and Appropriating It for Capitalism*. Stanford University Press. doi:10.1515/9781503609754
- [16] Nick Couldry and Ulises A. Mejias. 2019. Data Colonialism: Rethinking Big Data's Relation to the Contemporary Subject. *Television & New Media* 20, 4 (2019), 336–349. doi:10.1177/1527476418796632
- [17] Roger Dingledine, Nick Mathewson, and Paul F. Syverson. 2004. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium*. USENIX Association, 303–320. <https://www.usenix.org/conference/13th-usenix-security-symposium/tor-second-generation-onion-router>
- [18] Cory Doctorow. 2023. *Social Quitting*. Locus Magazine. <https://locusmag.com/feature/commentary-cory-doctorow-social-quitting/>
- [19] Cory Doctorow. 2025. *Enshittification: Why everything suddenly got worse and what to do about it*. Verso. <https://www.versobooks.com/products/3341-enshittification>
- [20] Arturo Escobar. 2018. *Designs for the pluriverse: Radical interdependence, autonomy, and the making of worlds*. Duke University Press. doi:10.1215/9780822371816
- [21] European Parliament and The Council of the European Union. 2022. Regulation (EU) 2022/1925 of the European Parliament and of the Council of 14 September 2022 on contestable and fair markets in the digital sector and amending Directives (EU) 2019/1937 and (EU) 2020/1828 (Digital Markets Act). 66 pages. <https://eur-lex.europa.eu/eli/reg/2022/1925/oj>
- [22] J. Nathan Foster, Michael B. Greenwald, Jonathan T. Moore, Benjamin C. Pierce, and Alan Schmitt. 2005. Combinators for bi-directional tree transformations: a linguistic approach to the view update problem. In *Proceedings of the 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '05)*. ACM, 233–246. doi:10.1145/1040305.1040325
- [23] Danny Goodman. 1998. *The Complete HyperCard 2.2 Handbook*. iUniverse. <https://www.iuniverse.com/en/bookstore/bookdetails/111054-The-Complete-HyperCard-2-2-Handbook>
- [24] Jay Graber. 2023. *Composable Moderation*. Bluesky. <https://bsky.social/about/blog/4-13-2023-moderation>
- [25] Colin M. Gray, Yubo Kou, Bryan Battles, Joseph Hoggatt, and Austin L. Toombs. 2018. The Dark (Patterns) Side of UX Design. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, 1–14. doi:10.1145/3173574.3174108
- [26] Carla F. Griggio, Joanna McGrenere, and Wendy E. Mackay. 2019. Customizations and Expression Breakdowns in Ecosystems of Communication Apps. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (11 2019), 1–26. doi:10.1145/3359128
- [27] Jens Emil Grønbaek, Marcel Borowski, Eve Hoggan, Wendy E. Mackay, Michel Beaudouin-Lafon, and Clemens N. Klokmoose. 2023. Mirrorverse: Live tailoring of video conferencing interfaces. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*. ACM, 1–14. doi:10.1145/3586183.3606767
- [28] H3 Contributors. [n. d.]. *H3: A Hexagonal Hierarchical Geospatial Indexing System*. <https://h3geo.org/>
- [29] Oliver L. Haimson, Daniel Delmonaco, Peipei Nie, and Andrea Wegner. 2021. Disproportionate Removals and Differing Content Moderation Experiences for Conservative, Transgender, and Black Social Media Users: Marginalization and Moderation Gray Areas. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (10 2021), 1–35. doi:10.1145/3479610
- [30] Mordechai Haklay and Patrick Weber. 2008. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing* 7, 4 (2008), 12–18. doi:10.1109/MPRV.2008.80
- [31] Aaron Halfaker and R. Stuart Geiger. 2020. ORES: Lowering Barriers with Participatory Machine Learning in Wikipedia. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW2 (10 2020), 1–37. doi:10.1145/3415219
- [32] Aaron Halfaker, R. Stuart Geiger, Jonathan T. Morgan, and John Riedl. 2012. The Rise and Decline of an Open Collaboration System: How Wikipedia's Reaction to Popularity Is Causing Its Decline. *American Behavioral Scientist* 57, 5 (12 2012), 664–688. doi:10.1177/0002764212469365
- [33] Theia Henderson, David R. Karger, and David D. Clark. 2025. Graffiti: Enabling an Ecosystem of Personalized and Interoperable Social Applications. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. ACM, 1–21. doi:10.1145/3746059.3747627
- [34] Benjamin Mako Hill, Andrés Monroy-Hernández, and Kristina Olson. 2010. Responses to remixing on a social media sharing website. In *Proceedings of the International AAAI Conference on Web and Social Media*. 74–81. doi:10.1609/ICWSM.V4I1.14028
- [35] Albert O. Hirschman. 1970. *Exit, Voice, and Loyalty: Responses to Decline in Firms, Organizations, and States*. Harvard University Press. <https://www.hup.harvard.edu/books/9780674276604>
- [36] Farnaz Jahanbakhsh, Amy X. Zhang, and David R. Karger. 2022. Leveraging Structured Trusted-Peer Assessments to Combat Misinformation. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (11 2022), 1–40. doi:10.1145/355637
- [37] Kylie Jarrett. 2014. The Relevance of "Women's Work": Social Reproduction and Immaterial Labor in Digital Media. *Television & New Media* 15, 1 (1 2014), 14–29. doi:10.1177/1527476413487607
- [38] Shagun Jhaver, Alice Qian Zhang, Quan Ze Chen, Nikhila Natarajan, Ruotong Wang, and Amy X. Zhang. 2023. Personalizing Content Moderation on Social Media: User Perspectives on Moderation Choices, Interface Design, and Labor. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW2 (9 2023), 1–33. doi:10.1145/3610080
- [39] Harry A. Kalodner, Miles Carlsten, Paul M. Ellenbogen, Joseph Bonneau, and Arvind Narayanan. 2015. An Empirical Study of Namecoin and Lessons for Decentralized Namespace Design. In *Proceedings of the 14th Workshop on the Economics of Information Security (WEIS '15)*. 1–23. <https://www.cs.princeton.edu/~arvindn/publications/namespaces.pdf>
- [40] Alan C. Kay. 1996. The Early History of Smalltalk. In *History of Programming Languages—II*, Thomas J. Bergin, Jr. and Richard G. Gibson, Jr. (Eds.). ACM, 511–598. doi:10.1145/234286.1057828
- [41] Royce Kimmons. 2011. Understanding Collaboration in Wikipedia. *First Monday* 16, 12 (12 2011). doi:10.5210/fm.v16i12.3613
- [42] Martin Kleppmann, Paul Frazee, Jake Gold, Jay Graber, Daniel Holmgren, Devin Ivy, Jeromy Johnson, Bryan Newbold, and Jaz Volpert. 2024. Bluesky and the AT Protocol: Usable Decentralized Social Media. In *Proceedings of the ACM CoNEXT-2024 Workshop on the Decentralization of the Internet (CoNEXT '24)*. ACM, 1–7. doi:10.1145/3694809.3700740
- [43] Clemens N. Klokmoose, James R. Eagan, Siemen Baader, Wendy Mackay, and Michel Beaudouin-Lafon. 2015. Webstrates: Shareable Dynamic Media. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, 280–290. doi:10.1145/2807442.2807446
- [44] Clemens Nylandsted Klokmoose, James R. Eagan, and Peter van Hardenberg. 2024. MyWebstrates: Webstrates as Local-first Software. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24)*. ACM, 1–12. doi:10.1145/3654777.3676445
- [45] Christine Lemmer-Webber, Jessica Tallon, Erin Shepherd, Amy Guy, and Evan Prodromou. 2018. *ActivityPub*. Recommendation. W3C. <https://www.w3.org/>

- TR/2018/REC-activitypub-20180123/
- [46] Karen Levy. 2023. *Data Driven: Truckers, Technology, and the New Workplace Surveillance*. Princeton University Press. doi:10.2307/j.ctv2ngx56j
 - [47] Geoffrey Litt, Josh Horowitz, Peter van Hardenberg, and Todd Matthews. 2025. *Malleable Software: Restoring User Agency in a World of Locked-Down Apps*. Ink & Switch. <https://www.inkandswitch.com/essay/malleable-software/>
 - [48] Geoffrey Litt, Paul Sonnentag, Max Schöning, Adam Wiggins, Peter van Hardenberg, and Orion Henry. 2024. *Patchwork: Version Control for Everything*. Ink & Switch. <https://www.inkandswitch.com/patchwork/notebook/2024-version-control/>
 - [49] Living Wage Institute. [n. d.]. *Living Wage Calculator*. <https://livingwage.mit.edu/>
 - [50] Wendy E. Mackay. 1991. Triggers and barriers to customizing software. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '91)*. ACM, 153–160. doi:10.1145/108844.108867
 - [51] Kaitlin Mahar, Amy X. Zhang, and David Karger. 2018. Squadbox: A Tool to Combat Email Harassment Using Friendsourced Moderation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, 1–13. doi:10.1145/3173574.3174160
 - [52] Alice E. Marwick and danah boyd. 2010. I tweet honestly, I tweet passionately: Twitter users, context collapse, and the imagined audience. *New Media & Society* 13, 1 (7 2010), 114–133. doi:10.1177/1461444810365313
 - [53] Ariadna Matamoros-Fernández. 2017. Platformed Racism: The Mediation and Circulation of an Australian Race-Based Controversy on Twitter, Facebook and YouTube. *Information, Communication & Society* 20, 6 (6 2017), 930–946. doi:10.1080/1369118X.2017.1293130
 - [54] Arunesh Mathur, Gunes Acar, Michael J. Friedman, Eli Lucherini, Jonathan Mayer, Marshini Chetty, and Arvind Narayanan. 2019. Dark Patterns at Scale: Findings from a Crawl of 11K Shopping Websites. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (11 2019), 1–32. doi:10.1145/3359183
 - [55] MDN contributors. 2025. *HTMLFrameElement: sandbox property*. MDN Web Docs. <https://developer.mozilla.org/en-US/docs/Web/API/HTMLFrameElement/sandbox>
 - [56] Ulises A. Mejias and Nick Couldry. 2024. *Data Grab: The New Colonialism of Big Tech and How to Fight Back*. University of Chicago Press. <https://press.uchicago.edu/ucp/books/book/chicago/D/bo216184200.html>
 - [57] Bryan Min, Allen Chen, Yining Cao, and Haijun Xia. 2025. Malleable Overview-Detail Interfaces. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. ACM, 1–25. doi:10.1145/3706598.3714164
 - [58] P. Mockapetris. 1987. *Domain Names—Concepts and Facilities*. RFC 1034. RFC Editor. doi:10.17487/RFC1034
 - [59] Jonathan T. Morgan, Siko Bouterse, Heather Walls, and Sarah Stierch. 2013. Tea and sympathy: crafting positive new user experiences on Wikipedia. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW '13)*. ACM, 839–848. doi:10.1145/2441776.2441871
 - [60] James Muldoon, Mark Graham, and Callum Cant. 2024. *Feeding the Machine: The Hidden Human Labour Powering AI*. Canongate Books. <https://canongate.co.uk/books/5234-feeding-the-machine-the-hidden-human-labour-powering-ai/>
 - [61] James Muldoon and Paul Raekstad. 2023. Algorithmic domination in the gig economy. *European Journal of Political Theory* 22, 4 (2023), 587–607. doi:10.1177/14748851221082078
 - [62] Theodore H. Nelson. 1974. *Computer Lib/Dream Machines*.
 - [63] Midas Nouwens, Rolf Bagge, Janus Bager Kristensen, and Clemens Nylandsted Klokmose. 2022. Consent-O-Matic: Automatically Answering Consent Pop-ups Using Adversarial Interoperability. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems (CHI EA '22)*. ACM, 1–7. doi:10.1145/3491101.3519683
 - [64] Elinor Ostrom. 2007. A Diagnostic Approach for Going Beyond Panaceas. *Proceedings of the National Academy of Sciences* 104, 39 (9 2007), 15181–15187. doi:10.1073/pnas.0702288104
 - [65] Evangelos Papadimitropoulos. 2021. Platform Capitalism, Platform Cooperativism, and the Commons. *Rethinking Marxism* 33, 2 (4 2021), 246–262. doi:10.1080/08935696.2021.1893108
 - [66] Santiago Perez De Rosso and Daniel Jackson. 2016. Purposes, concepts, misfits, and a redesign of Git. In *Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA '16)*. ACM, 292–310. doi:10.1145/2983990.2984018
 - [67] David Pierce. 2026. *You Can Make an App for That*. The Verge. <https://www.theverge.com/tech/928905/vibe-code-personal-software-revolution>
 - [68] Martin Potthast, Benno Stein, and Robert Gerling. 2008. Automatic vandalism detection in Wikipedia. In *Advances in Information Retrieval*. Springer, 663–668. doi:10.1007/978-3-540-78646-7_75
 - [69] Reid Friedhorsky and Loren Terveen. 2011. Wiki grows up: arbitrary data models, access control, and beyond. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration (WikiSym '11)*. ACM, 63–71. doi:10.1145/2038558.2038570
 - [70] Roman Rädle, Midas Nouwens, Kristian Antonsen, James R. Eagan, and Clemens N. Klokmose. 2017. Codestates: Literate computing with webstrates. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, 715–725. doi:10.1145/3126594.3126642
 - [71] Chand Rajendra-Nicolucci, Michael Sugarman, Ethan Zuckerman, and Rebecca Curran. 2023. *The Three-Legged Stool: A Manifesto for a Smaller, Denser Internet*. Initiative for Digital Public Infrastructure. <https://publicinfrastructure.org/2023/03/29/the-three-legged-stool/>
 - [72] Joseph Reagle. 2010. *Good Faith Collaboration: The Culture of Wikipedia*. The MIT Press. doi:10.7551/mitpress/8051.001.0001
 - [73] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. 2009. Scratch: programming for all. *Commun. ACM* 52, 11 (11 2009), 60–67. doi:10.1145/1592761.1592779
 - [74] Joost Rietveld and Melissa A. Schilling. 2021. Platform Competition: A Systematic and Interdisciplinary Review of the Literature. *Journal of Management* 47, 6 (7 2021), 1528–1563. doi:10.1177/0149206320969791
 - [75] Kevin Roose. 2025. *Not a Coder? With A.I., Just Having an Idea Can Be Enough*. The New York Times. <https://www.nytimes.com/2025/02/27/technology/personaltech/vibecoding-ai-software-programming.html>
 - [76] Yoel Roth and Samantha Lai. 2024. Securing federated platforms: collective risks and responses. *Journal of Online Trust and Safety* 2, 2 (2 2024). doi:10.54501/jots.v2i2.171
 - [77] Charlie F. Ruan, Yucheng Qin, Akaash R. Parthasarathy, Xun Zhou, Ruihang Lai, Hongyi Jin, Yixin Dong, Bohan Hou, Meng-Shiun Yu, Yiyan Zhai, Sudeep Agarwal, Hangrui Cao, Siyuan Feng, and Tianqi Chen. 2024. WebLLM: A High-Performance In-Browser LLM Inference Engine. *arXiv preprint* (2024). Version 2, revised April 2026. doi:10.48550/arXiv.2412.15803
 - [78] Jerome H. Saltzer and Michael D. Schroeder. 1975. The Protection of Information in Computer Systems. *Proc. IEEE* 63, 9 (9 1975), 1278–1308. doi:10.1109/PROC.1975.9939
 - [79] Nathan Schneider. 2024. *Governable Spaces: Democratic Design for Online Life*. University of California Press. doi:10.1525/luminos.181
 - [80] Nathan Schneider, Primavera De Filippi, Seth Frey, Joshua Z. Tan, and Amy X. Zhang. 2021. Modular Politics: Toward a Governance Layer for Online Communities. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (4 2021), 1–26. doi:10.1145/3449090
 - [81] Trebor Scholz. 2016. *Platform Cooperativism: Challenging the Corporate Sharing Economy*. Technical Report. Rosa Luxemburg Stiftung. https://rosalux.org.br/wp-content/uploads/2016/06/scholz_platformcooperativism_2016.pdf
 - [82] David A. Smith, Alan Kay, Andreas Raab, and David P. Reed. 2003. Croquet: A Collaboration System Architecture. In *First Conference on Creating, Connecting and Collaborating Through Computing, 2003. C5 2003. Proceedings*. IEEE, 2–9. doi:10.1109/C5.2003.1222325
 - [83] Nicholas Srnicek. 2016. *Platform Capitalism*. Polity. <https://www.wiley-vch.de/en/areas-interest/humanities-social-sciences/philosophy-12pl/continental-philosophy-12pl1/critical-theory-12pl13/platform-capitalism-978-1-5095-0487-9>
 - [84] Róbert Sumi, Taha Yasseri, András Rung, András Kornai, and János Kertész. 2011. Edit Wars in Wikipedia. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*. IEEE, 724–727. doi:10.1109/PASSAT/SocialCom.2011.47
 - [85] Imrat Verhoeven, Tamara Metze, and Ted van de Wijdeven. 2014. Do-ocracy’s democratic anchorage. In *ECPR Joint Sessions of Workshops 2014*. European Consortium for Political Research. Paper for the joint session on Systematising Comparison of Democratic Innovations: Advanced explanations of the emergence, sustenance and failure of participatory institutions. <https://hdl.handle.net/11245/1.442966>
 - [86] Daniela Vorkel. 2024. *WebSim.AI: No-Code Revolution — Weaving Immersive Web Experiences with AI-Powered Simulations*. Medium blog post. <https://medium.com/@daniela.vorkel/websim-ai-no-code-revolution-weaving-immersive-web-experiences-with-ai-powered-simulations-39d869a7aaee>
 - [87] Claudia Wagner, David Garcia, Mohsen Jadidi, and Markus Strohmaier. 2015. It’s a Man’s Wikipedia? Assessing Gender Inequality in an Online Encyclopedia. *Proceedings of the International AAAI Conference on Web and Social Media* 9, 1 (5 2015), 454–463. doi:10.1609/icwsm.v9i1.14628
 - [88] Wikipedia contributors. 2026. *Wikipedia:BOLD, revert, discuss cycle*. https://en.wikipedia.org/wiki/Wikipedia:BOLD_revert_discuss_cycle
 - [89] Amy X. Zhang, Grant Hugh, and Michael S. Bernstein. 2020. PolicyKit: Building governance in online communities. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. ACM, 365–378. doi:10.1145/3379337.3415858
 - [90] Ethan Zuckerman. 2020. *The Case for Digital Public Infrastructure*. Knight First Amendment Institute. <https://knightcolumbia.org/content/the-case-for-digital-public-infrastructure>