Internet Histories
Digital Technology, Culture and Society

ISSN: 2470-1475 (Print) 2470-1483 (Online) Journal homepage: www.tandfonline.com/journals/rint20

The loyal client: lessons from the past for the future of social media

Spencer Lane & Ethan Zuckerman

To cite this article: Spencer Lane & Ethan Zuckerman (09 May 2026): The loyal client: lessons from the past for the future of social media, Internet Histories, DOI: 10.1080/24701475.2026.2662718

To link to this article: https://doi.org/10.1080/24701475.2026.2662718

Published online: 09 May 2026.

INTERNET HISTORIES
https://doi.org/10.1080/24701475.2026.2662718

Routledge
Taylor & Francis Group

The loyal client: lessons from the past for the future of social media

Spencer Lane and Ethan Zuckerman

Manning College of Information and Computer Sciences, University of Massachusetts Amherst, Amherst, Massachusetts, USA

ABSTRACT

The client that is used to connect to a server serves a function that is almost as important as the server itself. The client governs how a user is able to interact with a digital system, the actions they are able to take, what information they can view, and what information is being shared. This paper presents a vision of a Loyal Client, a client that is aligned with the preferences of the end user rather than those of the platform. This is an old idea from the early days of the internet which we feel should be modernised for social media. We argue that a loyal client is a key piece of a generative internet and introduce four principles by which to evaluate whether a client is loyal, arguing that a loyal client should be: protective, interoperable, customisable, and usable. We explore this concept using three historical case studies: Usenet, instant messaging, and web browsers, connecting each case back to these principles.

ARTICLE HISTORY
Received 16 December 2025
Revised 10 April 2026
Accepted 16 April 2026

KEYWORDS
Loyal client; Usenet; instant messaging; web browsers; social media

1. Introduction

In the early days of the web, users primarily accessed networked services through programs that protected their interests - *loyal* clients (Rajendra-Nicolucci et al., 2023). With the rise of the smartphone, the app model has become dominant and services are accessed through restrictive dedicated mobile applications - *disloyal* clients. Even when accessing services on a desktop, many platforms provide a client similar to a mobile app, created in users' browsers using JavaScript. Modern social media platforms such as Facebook and X/Twitter limit user interactions outside of these restrictive clients with little user control.

Whereas the loyal clients of the early internet were integral to its growth and development, these new disloyal clients override user preferences, discourage innovation, and produce unhealthy platforms. For users whose primary internet experience has been mediated by smartphones and web apps, the concept of a flexible and loyal client might seem alien. An exploration of the rich history of clients highlights this powerful framework for evaluating existing and future clients: loyalty to user interests.

CONTACT Spencer Lane sdlane@umass.edu University of Massachusetts Amherst, Amherst, MA, USA.

Client-server architecture is a key part of the structure of the internet. The server acts as a central repository where multiple users fetch data using clients. Clients communicate with the server through one of many protocols, standardised rules for data transfer data.

Zittrain describes generativity[1] as "a system's capacity to produce unanticipated change through unfiltered contributions from broad and varied audiences.", this property has been essential to the growth and development of the internet (Zittrain, 2008a). Loyal clients are essential to generative systems - they enable experimentation through the creation of diverse clients. Bulletin Board Systems (BBSs) allowed users to dial in using a terminal protocol. Users connected through one of many clients and accessed software on the central computer alongside others. BBSs could be configured with different software and people created applications familiar to modern users including chat rooms, forums, and multiplayer games. One such game, the Multi-user dungeon (MUD), spawned many spinoffs and specialised clients that built additional functionality on the generic protocol. Some clients allowed users to write scripts that automated character behaviour, such as automatically healing a wounded ally. According to one MUD designer, "the third-party tools were seen as something that added invaluable interface enhancements" (Koster, 2008). None of this innovation would have been possible if the MUDs required specific clients to connect. Loyal telnet clients enabled a generative ecosystem of MUDs and innovation for both the platforms and the clients.

In this paper, we explore the history of the loyal client in detail. First we introduce the properties of loyal clients and open servers. Then we examine these properties through three case studies: Usenet, Instant Messaging, and Web Browsers.

2. Loyal clients, open servers

2.1. Loyal clients

A loyal client's allegiance must be to the user and their interests. Websites have monetary incentives to serve intrusive ads. Modern browsers protect users by blocking pop-ups windows and allowing the installation of ad-blocker extensions. Despite legal challenges, courts have upheld the right to ad-blockers (AdGuard, 2025). The browser is loyal to the user in the zero-sum game of whether advertisements are shown. The expectation of these options holds even when the company producing the browser has a financial interest in serving ads, such as the case of Google Chrome. Loyalty is a spectrum, but a good heuristic is how well the client embodies four key principles: a loyal client should be protective, customisable, interoperable, and usable.

Here, we explore these principles through comparing web browsers and email clients with Facebook. Web browsers act as a client for websites, interacting using a protocol such as hyper text transfer protocol (http). They retrieve the mark-up files, JavaScript programs, and other data needed to render each site. Email clients retrieve email data and attachments using an email protocol such as Post Office Protocol version 3 (POP3) and send data using a protocol such as Simple Mail Transfer Protocol (SMTP). Facebook is instead accessed using a mobile application built by Meta or by a JavaScript application within the user's browser. These apps form clients that are

tightly coupled to Facebook's servers with most the important computation performed on Meta's servers.

A loyal client is *protective*; it guards the user's interests over competing interests. Web browsers generally have privacy settings which let users choose how their data is shared. Conversely, Facebook provides minimal options for users to control its collection of fine grained app usage data.

A loyal client is *customisable*; it provides users with a high degree of control over the selection and formatting of content. Most email clients provide a wealth of filtering and display options. Facebook on the other hand allows users very limited control over what appears in their feeds, even directly inserting unblockable ads.

A loyal client is *interoperable*; it provides the user with the ability to choose different service providers and to switch providers if they desire. Most email clients can handle multiple accounts from multiple providers and can send messages to any address. Meanwhile, the only supported method of sending a Facebook post outside Meta's ecosystem requires sharing a link back to Facebook - and it only works for fully public posts.

A loyal client is *usable*; it facilitates the user's goals, rather than frustrate them with predatory design and dark patterns (Gray et al., 2018). As different users will have different use cases, usability depends on an environment that allows for experimentation and the creation of clients with varying affordances.

The central property of a loyal client is that when the interests of the user and the interests of the server are in conflict, a loyal client sides with the user.

2.2. Open servers

Using a loyal client requires relatively unrestricted access to user data. This is generally done through the platform providing an open protocol or an Application Programming Interface (API). Protocols allow clients to connect to more than one provider. Internet Relay Chat (IRC) (Kalt, 2000) is a widely used open protocol supported by dozens of different chat servers. Such open standards are often maintained by groups of technologists operating independently of any one company. An open protocol allows different kinds of clients to connect to the same platform and for clients to connect to different platforms that use the same protocol.

APIs allow different kinds of clients to connect to the same server, but that API is often only provided by one platform. Consider Reddit, which once boasted a diverse set of clients such as Luna (compatible with screen readers for the visually impaired) (Smith, 2022). By providing a robust API, Reddit shared the load of providing clients to meet the needs of diverse communities with developers experienced in serving those communities. When Reddit drastically raised the price of accessing its API, some developers stopped maintaining their clients because they could no longer afford access (iamthatis, 2023). API changes can be performed unilaterally by platforms which makes this approach to interoperability more fragile than open protocols.

Even when a server does not enable interoperability, it's sometimes possible to access data through a process that Cory Doctorow termed "adversarial interoperability" (Doctorow, 2019). In adversarial interoperability, developers force a closed protocol open. Microsoft made documents created with its word processor inaccessible to

other platforms by using a proprietary file format, .doc. Sun Systems and Open Office reverse-engineered the file format and allowed users to read and write .doc files despite Microsoft's objections, and eventually Microsoft released it publicly (Microsoft, 2008; Rentz, 2007). However, companies like Meta make it very difficult - technically and legally - to engage in this sort of adversarial interoperability.

An open server either uses an open protocol that a client can easily interoperate with, or provides a robust, well-supported API that is affordable for most software developers to use. An open server needs to be agnostic about the clients that connect to it, providing the same data to both first party and third party clients.

3. Case study: Usenet

3.1. Usenet overview

Usenet is a social media ecosystem predating the web in which users share articles to news servers which store those messages and forward them to other nodes as they connected. At one time it was considered the largest interconnected computer network. It has always operated under an open protocol (UUCP or NNTP (McIlroy, 1971; Kantor & Lapsley, 1986)) and users have always had a choice of newsreader clients with which to download articles from one or more user-selected news servers.

Usenet messages are hierarchically organised into newsgroups like "sci.math" or "talk.atheism." Originally the top level categories were fa.* (Arpanet derived groups), mod.* (moderated), and net.* (unmoderated). In 1987 during "The Great Renaming.", they were reorganised into: comp.*, misc.*, news.*, rec.*, sci.*, soc.*, and talk.*(Hardy, 1993; Lee, 2002). In 1988 an alternative hierarchy (appropriately called alt.*) was created. Because of Usenet's open protocol (and loyal clients which allowed users to choose newsgroups and providers), these hierarchies existed in parallel (Hardy, 1993).

Usenet had an end-to-end system design, with processing primarily at the end of the network chain. Administrators of machines passing Usenet messages exerted little control over which messages were passed, but chose which hierarchy subsets to distribute (Pfaffenberger, 1996; Resnick et al., 1994). Users configured their newsreaders to filter out unwanted messages. These factors make Usenet a good case study of an early social media ecosystem involving a flexible protocol and loyal clients.

3.2. The Backbone Cabal and the alt hierarchy

The Backbone Cabal was a semi-formal group of system administrators who operated important routeing nodes and passed a plurality of Usenet traffic. They collaborated to institute policies, some *via* software and some by agreeing to restrict which newsgroups their systems distributed. Every Usenet article specifies the newsgroup(s) it is intended for. It is processed by the server it was submitted to and forwarded to other servers. Those servers process the message and decide whether to pass it on. If major servers refused to carry a certain message or certain kinds of traffic, it was difficult if not impossible for that message to spread. This gave the administrators of high-traffic servers significant power. As administrators began to coordinate their actions, they were termed - and eventually self-identified as - a cabal (Hardy, 1993).

Usenet administrators, including members of the cabal, were responsible for many network processes, including the process for instantiating a new newsgroup, white listing it as valid traffic. In order for new groups to propagate, many administrators would need to make these changes. Proposals for the creation of new groups went through a voting process. When a newsgroup creation proposal vote succeeded, it was up to administrators whether their systems would forward the new group. In several cases, the vote succeeded, but the Backbone Cabal refused to carry the group. Many administrators, unhappy with this refusal, took action, creating the alt hierarchy, a set of Usenet groups for topics that did not fit well within the existing taxonomy (Lee, 2002).

The creation of newsgroups in the alt hierarchy relied on the goodwill of a subset of administrators. When creating alt connections, they agreed "to pass all alt traffic to each other and to nurse the net along" (Hardy, 1993). This resilient routeing (and the fact newsreader users could change providers) enabled this digital coup, a generative change to the network that its most powerful members opposed.

Even after the Backbone Cabal waned, the network remained resilient to censorship by administrators. New routes could be created to pass the alt hierarchy around sites that refused to carry it. This sometimes permitted dissemination of unapproved groups. If users sent messages to arbitrary newsgroup addresses some sites might pass them along without the proper voting procedures. Though some administrators maintained a list of "bogus newsgroups" (implying sites should not forward messages addressed to these groups), others were happy to forward these messages (Lee, 2002).

The emergence of the alt hierarchy against the wishes of the Backbone Cabal demonstrates the generative nature of open protocols and interoperable clients. In a world where a small group (or cabal) controlled both servers and clients, it would have been impossible to introduce new groups like the alt hierarchy. But because Usenet was engineered around loyal clients and multiple servers, users could choose whether they wanted to see these new alt newsgroups or filter them out. Users and administrators did not need to modify the software in order to start using new groups - instead users added alt newsgroups to their subscriptions and administrators added alt to the list of newsgroups their systems retrieved and forwarded.

3.3. Newsreaders and spam

On April 12th, 1994, numerous newsgroups received a pitch from two lawyers offering assistance registering for a Greencard lottery (Campbell, 1994). Those lawyers had sent the first commercial spam message on Usenet. It was not the first unwanted Usenet message. In 1993 one user set up a program that posted a long article denying the Armenian Genocide in response to any instance of the word "Turkey" … including recipes for Thanksgiving dinner (DeVoto, 1994). Many users hated this bot, but Usenet admins collectively decided to allow users to filter out spam within their clients rather than preventing content from spreading through the network entirely. Instead of unwanted speech being stopped at the server where it originated, users maintained a killfile which told their client not to display content meeting various criteria (Free Software Foundation, 2025; Gaffin & Heitkotter, 1994).

As Usenet grew, so did commercial spam. The policy of neutrality towards traffic content was largely maintained but other spam filtering systems were created. Administrators used server-side programs to help prevent their sites from spreading spam. These were often based on heuristics such as the Breidbart Index, which measured how widely a post is crossposted (Lewis, 1998). Because newsreaders can connect to arbitrary providers, if an administrator decides to make a change that eliminates content users wanted, they can choose another provider.

3.4. Usenet conclusions

Usenet's history highlights how the structure of platforms impacts user ownership of data and the ability to create loyal clients. It also illustrates many trade-offs in systems that are compatible with loyal clients and how this system model interacts with the loyal client principles.

Usenet provides an example of *protective* client, at least when it comes to spam and abusive behaviours. Usenets decentralised design made it difficult to permanently remove users or content. Instead, users were empowered to process and remove wanted posts with software on their own computer. On the other hand, the inability to remove or edit Usenet posts means users could not revoke their posts, and abusive content could not be easily removed from the network. While users and administrators could issue cancel messages there is no guarantee that other sites will respect those cancellations. Similar problems persist in modern decentralised systems. Newsreaders were entirely responsible for protectiveness, whereas a centralised system can exercise control server-side.

The alt hierarchy story shows the utility of *interoperable* systems. Interoperability can lead to a generative environment and to systems evolving organically based on community desires. The ability to change providers within loyal newsreaders allowed Usenet to grow in ways powerful players opposed. The inherent interoperability provided by Usenet's protocol allowed the network to adapt.

Usenet evolved during a period of emerging networks connecting small numbers of computers rather than our modern, vastly interconnected global network. Its radically end-to-end architecture allowed clients to be *customisable*, but its developers also presumed sophisticated users capable of setting up and managing individual filters and killfiles - an imperfect analogy to our vast, highly accessible social media environment.

Usenet's open structure allows clients to experiment with different affordances and designs, increasing the likelihood that a user could find a *usable* client. One of the techniques for performing content recommendation is called collaborative filtering. It was pioneered on Usenet by researchers who created the Grouplens newsreader (Resnick et al., 1994). The ability of users to select their own Usenet client enabled the creation of this technology that powers many modern recommendation systems.

4. Case study: Chat

4.1. Chat overview

Chat - synchronous real-time text based messaging between two or more parties - is one of the most common forms of communication on the internet and has been

part of the internet landscape since at least the 1970s (Latzko-Toth, 2010). The earliest chat systems enabled terminal-to-terminal conversations between users connected to the same shared server. Later systems allowed messages to be sent *via* the internet between users on separate servers. Chat rooms, which became popular in 1988 with the rise of IRC, allowed groups of users around the world to connect in a shared "space", referred to as a "channel".

While IRC still exists, it was supplanted by proprietary options (WhatsApp, Discord, etc). Most contemporary chat services have one provider and one client, though Meta has been ordered by the European Union (EU) to enable interoperability for its services (Meta Platforms Inc, 2024).

Many chat systems are already designed with aspects of the loyal client in mind. The Matrix protocol is a distributed protocol that supports modern chat features like end-to-end encryption (E2EE) (The Matrix.org Foundation CIC, 2025). A variety of Matrix clients have been developed (The Matrix.org Foundation CIC, n.d.-b) and various groups are working on "bridges" to connect with other chat services (The Matrix.org Foundation CIC, n.d.-a).

In this section, we explore three lessons from the history of chat systems: the conflict over interoperability between providers in 1999, an early protocol designed around interoperability, and the EU's regulation on interoperability.

4.2. The IM wars

In 1999, the chat landscape was dominated by Yahoo! Messenger, America Online (AOL) Instant Messenger (AIM), and ICQ with the latter two collectively hosting 80 million users (Hansell, 1999a). While older services like IRC pre-dated them, these services came bundled with internet access software or popular search portals and quickly gained larger market shares. When Microsoft developed their own competing service, Microsoft Network (MSN) Messenger Service, they reverse engineered AIM's proprietary Open System for Communication in Realtime (OSCAR). MSN Messenger launched with the ability to communicate with AIM users. AOL quickly modified their protocol to break compatibility and Microsoft quickly patched to restore it (Auerbach, 2014).

AOL sent cease and desist letters to Microsoft, who denied AIM's terms of service had been violated (Hansell, 1999b). Eventually AOL exploited a bug in the AIM client to verify whether a particular client was authentic (Chappell, 2008).

Later, Yahoo! and Prodigy also set up their chat systems to interoperate with AIM. Their companies enabled interoperability by using code from an open source client published by AOL itself (Hansell, 1999a). Despite operating a closed protocol that called itself open, and specifically working to shut down attempts at interoperability, a spokesperson for AOL stated the company believed instant messaging would eventually be fully interoperable (Hansell, 1999b).

4.3. Jabber/XMPP

While the IM wars raged, Jabber was an early instant messaging platform designed around a protocol. Jabber was dedicated to open and interoperable instant messaging.

Its release included a server and several clients - from the beginning users were empowered to choose a client suited to them (XMPP Standards Foundation, n.d.). The protocol was renamed the Extensible Messaging and Presence Protocol (XMPP) when it was submitted to the Internet Engineering Task Force (IETF) (though informally continued to be referred to as "Jabber"). The XMPP protocols were eventually formalised in Request for Comments (RFC) 3920 and 3921 (Saint-Andre, 2004a, Saint-Andre, 2004b).

As you would expect, users with an account on a server running XMPP are able to communicate with users on other servers running XMPP. XMPP users can also communicate with users on servers running completely different protocols. An XMPP client can handle all three types of communication[2] seamlessly and invisibly to its users.

XMPP launched in 1999, the height of the IM wars. Its vision of interoperability became the industry standard as each instant message provider discovered users want to communicate with friends across platforms. While contemporary companies like Facebook refused interoperability until forced by EU policymakers, early chat providers came to realise isolation wasn't beneficial and eventually either tolerated gateways between their protocols and XMPP, or otherwise embraced an interconnected ecosystem.

4.4. The digital markets act

The EU has recently begun regulating "Number-Independent Interpersonal Communications Services" (N-IICS) - for-profit chat applications independent of phone numbers. They are defined as part of the European Electronic Communications Code (EECC) and are also featured in the Digital Markets Act (DMA), a piece of EU legislation designed to regulate digital "gatekeeper" companies with a large and entrenched market position who operate one or more "core platforms services", a category which includes search engines, social networks, and N-IICSs. Article 7 requires N-IICSs operated by designated gatekeeper companies to provide tools for interoperability with other N-IICSs operating in the EU (The European Parliament and the Council of the European Union, 2022). This is a positive signal that the EU is open to using regulation for the benefit of internet users, but there are some complications.

First, the DMA is limited in its scope. There are only two N-IICSs currently regulated by the act, both of which are owned by Meta: Messenger and WhatsApp (European Commission, 2023). The regulation does not target services under certain market thresholds, even when operated by organisations denoted as gatekeepers under the DMA. This means the EU now requires Meta (and only Meta) to make its messaging services interoperable with competitors. To its credit, Meta announced it has made progress making WhatsApp interoperable and Messenger would follow soon after (Meta Platforms Inc, 2024). In November 2025 Meta announced that interoperability between WhatsApp and two partner services would begin deployment to users in Europe over the next few months (Meta Platforms Inc, 2025).

Another issue is both services targeted by the DMA incorporate E2EE. Meta announced they would be using the Signal protocol for doing E2EE between WhatsApp and partner services as it represents a "gold standard" (Meta Platforms Inc, 2024).

In Meta's initial implementation, the partner service's server will be primarily used for authentication and push notifications, while messages will be sent directly to clients *via* a persistent connection. The announcement also mentions the possibility of allowing partner services to place proxies between their clients and WhatsApp, potentially controlling what is exchanged *via* WhatsApp.

This could also remove the current restriction that the third party use the WhatsApp protocol within the client, creating something similar to an XMPP bridge. Meta's announcement notes this creates new security risks in that message metadata would be given to third party services and Meta would lose direct access to some of the metadata it currently uses for spam filtering (Meta Platforms Inc, 2024). Though the content of the messages is not visible, a user who is sending thousands of messages per second is probably a spammer who should be blocked.

This regulation appears to be pushing Meta in the right direction. They have outlined an approach to interoperability within their N-IICSs that maintains security guarantees as best they can while allowing third party services to interoperate. The fact the EU passed this consumer focused regulation is positive and there are groups working on similar regulatory frameworks for social media providers. Meta's moves towards making WhatsApp and Messenger interoperable suggest regulation can push platforms towards interoperability and even towards loyal clients.

4.5. Chat conclusions

Each of the above case studies explores different methods of making clients *interoperable*. MSN's quest to force interoperability with AOL is an illustration of adversarial interoperability. Even though AOL resisted interoperating with MSN Messenger, the Microsoft team reverse engineered the AIM protocol and interoperated anyway.

XMPP sometimes employed adversarial interoperability, building bridges to proprietary chat services whether licitly or otherwise. More importantly, unlike MSN's closed protocol, XMPP uses an open protocol allowing multiple clients, including loyal clients. The IM wars of the turn of the century were ultimately won, not by one particular protocol, but by systems designed to interoperate. This suggests users don't want to think about what network they need to be on, they just want to communicate.

The XMPP case also illustrates the power of middleware. A loyal client interfaces with a trusted server to send data to a separate trusted program, in this case in the form of a gateway. While many of these gateways are run on the server hosting the XMPP instance, this is not required architecturally.

As with the DMA, regulatory solutions can achieve interoperability in the case of resistance from providers. It is possible to achieve interoperability by combining technical, protocol, and regulatory means.

Each of these models of interoperability influences the availability of *protective*, *customisable*, and *usable* clients. Once made interoperable, users could choose the AIM or MSN client based on their preferences. XMPP gateways enabled an ecosystem of clients, letting users select clients which met their needs and aligned with their values. The EU mandate could open Meta's services to similar client development.

5. Case study: Web browsers

5.1. Browser overview

Web browsers are the quintessential example of a loyal client. Across the World Wide Web, servers are agnostic to which browser is being used to access them. Most browsers give a variety of options for customisation and to prevent websites from engaging in certain harmful behaviours. Browsers can be extended with third-party software, and many popular third-party extensions disable ads or tracking systems intended to benefit advertisers.

Critically, browsers are designed around a protocol, not a platform. The browser connects to any website if the correct IP address and port are known. Interoperability by default likely contributed to the rapid growth of the World Wide Web after its introduction. In this section, we will discuss the features that make modern browsers a paradigmatic form of loyal client and contrast these browsers with the walled gardens of the early internet.

5.2. Loyal browsers

From its earliest days, the web was designed to allow users to make choices about how content appeared. Through the 1990s, web designers had to consider users might have images turned off to conserve bandwidth or be using an entirely text-based browser. Modern browsers are still designed to be loyal to the user. Even browsers known to share user data, such as Google Chrome, provide settings allowing opting out of data collection (Google Inc., n.d.). Browser extensions allow users to add additional layers of privacy, control, and security between themselves and the server. This control is key because the desires of users and services are often in conflict.

The challenge has grown with the deployment of "third-party cookies", which work to track users across multiple websites, allowing data brokers to sell user profiles extrapolated from web activity. As tracking efforts got more aggressive, the EU responded by amending their e-Privacy directive, which requires websites to obtain user consent before adding cookies (The European Parliament and the Council of the European Union, 2002, 2009). Browsers already allowed users to turn off and remove cookies, and to install surveillance blocking extensions. Arguably, the ability to block cookies across all sites is a more elegant implementation of user choice than popups on every website.

Originally, web browsers could not accommodate complex multimedia interactions but early web designers wanted to use popular tools like Macromedia Flash and Realplayer. Early browser add-ons were helper applications maintained by third-party developers, used by the web browser to display unsupported content (but only if the user chose to install the extension). As the information delivered by webservers grew more complex, delivering full-featured applications like word processors and email clients within the browser window, browsers gained capabilities too.

All major desktop web browsers now provide marketplaces for extensions or plug-ins (Schiller, 2021). Similar to Flash or Javascript applications, browser extensions provide functionality and also allow the user to modify how they interact with

particular websites. For example, the ad blocker, which reduces the number of ads displayed to users, potentially reduces the revenue browsing would otherwise generate.

Extensions have also been used to extend the functionality of social media sites. Unfollow Everywhere was an extension that automated the process of unfollowing friends on Facebook (Barclay, 2021). New social features can also be created by extensions such as Trustnet, which displays user generated comments about whether particular pages contain misinformation or disinformation (Jahanbakhsh & Karger, 2024).

Browser extensions are an example of "middleware", a term proffered by Fukuyama and colleagues to describe "software, provided by a third party and integrated into the dominant platforms, that would curate and order the content users see.(Fukuyama et al., 2020)". Fukuyama sees middleware as a possible solution to contemporary internet problems, like mis- and disinformation or content likely to trigger body image issues. Because web browsers - since inception - have given users the ability to enforce their preferences they provide plenty of useful examples (Zuckerman & Brickman 2024).

5.3. Walled gardens

The contemporary web differs sharply from the consumer internet as envisioned by early business pioneers. Two 1990s companies worked to implement a vision of networked content in which their software would be central to all online interactions: AOL and Netscape.

In the early 1990s, AOL served millions of customers who dialled into their servers *via* modem. Users were recruited *via* more than 1 billion mail distributed CDs. The software on those CDs allowed users to connect to AOL, and AOL only (Edwards, 2015).

AOL initially believed it could provide all the services users might want, including a proprietary email system and AIM, but AOL's services increasingly took a back seat to the variety and diversity of information available on the World Wide Web (Jones, 2016). As other providers began offering faster and cheaper service, AOL looked for other revenue models and diversified into web content, using its valuable stock to purchase "old media" giant Time Warner, a transaction that was ultimately unsatisfactory.

Netscape initially looked like a challenger for dominance of the early web, before being purchased by AOL just prior to the Time Warner merger. When Netscape went public in 1995, it was seen as a milestone in financial markets: a company purely focused on the internet, challenging established tech players like Microsoft. While Netscape had some moneymaking products, its visibility was based on its free browser. Netscape needed to figure out how to keep its users loyal to their browser when Microsoft was including Internet Explorer with the ubiquitous Windows operating system. Netscape's answer was to create new functionality not easily duplicable.

Netscape's solution temporarily turned the web from a system united around a single protocol to one that behaved differently for different clients. Netscape created a new language - Javascript - allowing web page authors to add significant interactivity. This presented a dilemma for developers: using the language for cool new features in Netscape created pages incompatible with other web browsers.

Before Netscape was absorbed into AOL, it did something extremely important: it released its browser source code to a non-profit foundation, Mozilla, which continues to maintain the open source Firefox browser (Netscape Communications Corporation, 1998). Javascript was adopted across the industry and now serves as a standard for the contemporary web (Rauschmayer, 2014).

Both AOL and Netscape embraced visions of a walled garden, an attractive subset of the internet they controlled. AOL tried to keep their customers subscribed by building exclusive services like AIM, a chat client so popular AOL believed people would stay with the service to have access (Jones, 2016). After releasing the free AIM client, AOL was left with the failed gambit of offering Time Warner's "premium" content and discovered the audience was limited. Netscape's walled garden was geekier, offering capabilities unavailable elsewhere. Again, developers and users rebelled - a world where your browser determines what portions of the web are accessible is a miserable experience. Ultimately, Netscape's advances were embraced, but in a way that rejected the choice the company tried to force.

Both Netscape and AOL violated the central idea of the loyal client: the choice belongs to the user, and the user should be able to switch to the best tools anytime. Hoping to achieve business advantage, both companies ultimately ended up betraying trust and sacrificing relevance.

5.4. Web browser conclusions

Web browsers are a modern model of the loyal client familiar to most internet users. They also highlight the generative potential of loyal clients. While not every web surfer installs extensions, those that do are able to improve on the browser's ability to be protective, add customisation options, and modify the interfaces on websites to make them more usable. Users expect browsers to be interoperable. Attempts to break those expectations by creating walled gardens have historically failed.

The growth of modern social media roughly coincides with the rise of the smartphone. Many users primarily interact with web browsers through their smartphone, and many platforms push users towards their dedicated, single function phone applications. In both cases, these clients are less loyal than desktop web browsers, typically not allowing extensions and offering few privacy options. Even when accessing social media sites through a desktop browser, many are web apps that work to avoid the customisations and extensions users have installed. For example, some sites will attempt to circumvent ad blockers (Zhao et al., 2017) and other sites have been known to modify their code periodically to break plugins (Merrill & Tobin, 2019). Some platforms, such as Facebook, dedicate significant resources to these actions. Despite this, we learn from the examples of AOL and Netscape that loyalty to the user, at least in a domain where users expect it, pays off.

Though many browsers have had their loyalty reduced over the years, they remain a quintessential example of each of the four properties of a loyal client. They exhibit *protective* behaviour, blocking pop ups and providing options to help block tracking. Most are *customisable*, with extensions available to change browser operations. They

are inherently *interoperable*, allowing users to query any IP address using HTTP or HTTPS. The extensions also help make them *usable*, filling in accessibility features and other personalised interfaces not provided by the designers.

6. Discussion

The loyal client provides a conceptual framework for analysing past developments in digital communication mediums and suggests a hopeful model for the future of social media. Through each of the case studies discussed in this paper, we examine a struggle between an open and a closed paradigm, with the open paradigm serving as a catalyst for generative creation. In Usenet we see a story of a network growing beyond the desires of its active administrators because of its protocol. In chat we see three models of interoperability that ultimately lead to a robust ecosystem of loyal chat clients. In the web browser, we have an example of the extent to which an open and generative protocol can lead to rapid growth and a nearly limitless potential for future development.

These examples provide a vision of one possible future - but it is not a guarantee. Usenet ultimately declined in popularity as it was flooded with spam and new, friendlier alternatives emerged. Most online chat today occurs in WhatsApp, Slack, and Discord, platforms that lack the open nature and loyal clients of XMPP. Much of what would once be done through a browser is now frequently done through a browser app which lacks many elements of the loyal client, or in dedicated phone apps which fall far short of that ideal.

When we envision a loyal client for social media, it would have a lot in common with an email client. It would be *protective*, using feed generation algorithms that do not monopolise user attention. It would be *customisable*, providing options for the user to adjust and ideally allowing its services to be used with different user interfaces. It would be *interoperable*, pulling data from multiple services using something akin to XMPP bridges. It would be *usable*, making the user's goals easier to achieve, not standing in their way.

Building this client poses significant challenges. Gateways require maintenance and sites like Facebook have historically tried to break such software (Merrill & Tobin, 2019). Making a client simultaneously interoperable and usable is an difficult interface problem. Pulling data from platforms to send to loyal clients removes some actions that centralised platforms take to be protective, and there are questions about data ownership and privacy. Despite these challenges, we believe it is important to move towards creating such a client.

The history of the internet is full of struggles over whether various systems will be more closed or more open. While options for open systems exist, mainstream social media has currently swung very far towards the side of closed. One effect of that is the lack of loyal clients for social media. Conversely, if a loyal client were to be created, its intrinsic properties (protective, interoperable, customisable, usable) might be expected to push social media in a more open direction, empowering users and opening up the social media space to the kind of generative creativity that we see in these case studies.

The generativity of the web depends on three key pieces: open protocols, multiple competing servers, and a diversity of loyal clients. As predicted by Zittrain in The

Future of the Internet and How to Stop It (Zittrain, 2008b), the shift from the computer to the mobile phone has caused us to lose the loyal client in favour of the walled garden of competing apps. In the face of this labyrinth of walled gardens, we would do well to remember the model of the loyal client and to use it as a standard for what we should expect from the digital tools we use to interact with each other online.

The main lesson that we should take away from these case studies is that we need to centre user agency, not the interests of platforms, in design and enabling legislation. While we remain within the competing walled gardens of social platforms over which we have very little control, users will continue to be at a disadvantage and we will fail to develop the full potential of the technology. In the absence of better tools, many users, including the authors, will continue to use these walled gardens rather than abandon our place in our various social spheres. A loyal client for social media will help create an environment where it's possible to choose a better experience and build a healthier future.

Notes

1. This notion of generativity is unrelated to the usage of the term in the field of artificial intelligence as in "generative AI".
2. Messages within the same server, messages to a different server running XMPP, messages to a different server running a different protocol.

Acknowledgement(s)

We would like to thank Kimberly Lane for her dedicated support, internet savvy, and many, many conversations. We would also like to thank the members of our Loyal Client discussion group for feedback and many conversations. Finally, we would like to thank the entire team at iDPI for their support along the way.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was funded by the Ford Foundation and the MacArthur Foundation.

Notes on contributors

Spencer Lane holds a Ph.D. from the Manning College of Information and Computer Science at the University of Massachusetts Amherst and a member of the Initiative for Digital Public Infrastructure. He also holds an S.M. from the Massachusetts Institute of Technology and a B.S. from Old Dominion University. His doctoral research focused on creating a loyal client for social media and on tools to enable end user evaluation of filtering and ranking algorithms.

Ethan Zuckerman is an Associate Professor of Public Policy, Communication and Information at the University of Massachusetts Amherst and the founder of the Initiative for Digital Public Infrastructure.

ORCID

Spencer Lane http://orcid.org/0009-0000-8897-8123
Ethan Zuckerman http://orcid.org/0000-0002-5970-9116

References

AdGuard. (2025). *Revived anti-ad blocking lawsuit puts users' rights and the Internet's future at risk*. Retrieved December 10, 2025, from https://adguard.com/en/blog/axel-springer-adblock-plus-case-revived.html

Auerbach, D. (2014). *Chat wars*. Retrieved May 16, 2024, from https://www.nplusonemag.com/issue-19/essays/chat-wars/(Section: Essays)

Barclay, L. (2021). Facebook banned me for life because I help people use it less. *Slate*. Retrieved December 21, 2024, from https://slate.com/technology/2021/10/facebook-unfollow-everything-cease-desist.html

Campbell, K. (1994). *A NET.CONSPIRACY SO IMMENSE. Chat-ting With Martha Siegel*. Electronic Frontier Foundation. Retrieved December 6, 2024, from https://web.archive.org/web/20071125201904/http://w2.eff.org/legal/cases/Canter_Siegel/c-and-s_summary.article

Chappell, G. (2008). *America online exploits bug in own software*. Retrieved June 5, 2025, from https://www.geoffchappell.com/notes/security/aim/index.htm

DeVoto, J. A. E. (1994). *The Zumabot's Tale*. Retrieved December 6, 2024, from http://www.jaedworks.com/shoebox/zumabot.html

Doctorow, C. (2019). *Adversarial interoperability*. Retrieved December 6, 2024, from https://www.eff.org/deeplinks/2019/10/adversarial-interoperability

Edwards, P. (2015). *In memoriam: AOL CDs, history's greatest junk mail*. Retrieved June 17, 2025, from https://www.vox.com/2015/5/12/8594049/aol-free-trial-cds

European Commission. (2023). *Questions and Answers: Digital Markets Act: Ensuring fair and open digital markets* [Text]. Retrieved April 12, 2024, from https://ec.europa.eu/commission/presscorner/detail/en/qanda 20 2349

Free Software Foundation. (2025). *Kill files*. Retrieved November 28, 2025, from https://www.gnu.org/software/emacs/manual/html_node/gnus/Kill-Files.html

Fukuyama, F., Richman, B., Goel, A., Katz, R. R., Melamed, A. D., & Schaake,M. (2020). Middleware for dominant digital platforms: A technological solution to a threat to democracy. *Stanford Cyber Policy Center Freeman Spogli Institute*. https://cyberlaw.stanford.edu/content/files/s3fs-public/cpc-middlewareffv2.pdf

Gaffin, A., & Heitkotter, J. (1994). Killfiles - The cure for all that ails you. In *EFF's (Extended) guide to the internet* (pp. 77–79). Electronic Frontier Foundation. Retrieved June 5, 2025, from http://archive.org/details/B-001-004-387

Google Inc. (n.d.). *Turn "Do Not Track" on or off*. Retrieved November 28, 2025, from https://support.google.com/chrome/answer/2790761

Gray, C. M., Kou, Y., Battles, B., Hoggatt, J., & Toombs, A. L. (2018). *The Dark (Patterns) Side of UX Design* [Paper presentation]. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, Montreal, QC, Canada. April 1–14. ACM. Retrieved November 23, 2025, from https://dl.acm.org/doi/10.1145/3173574.3174108 https://doi.org/10.1145/3173574.3174108

Hansell, S. (1999a). In cyberspace, rivals skirmish for control over messaging. *The New York Times*. Retrieved December 10, 2025, from https://archive.nytimes.com/www.nytimes.com/library/tech/99/07/biztech/articles/24mail.html

Hansell, S. (1999b). Positions harden in instant-message fight. *The New York Times*. Retrieved December 10, 2025, from https://archive.nytimes.com/www.nytimes.com/library/tech/99/07/biztech/articles/28mail.html

Hardy, H. E. (1993). *History of the Net* [Master's thesis]. Grand Valley State University. Retrieved June 4, 2025, from https://devin.com/cruft/hardy.html

iamthatis. (2023). *Apollo will close down on June 30th. Reddit's recent decisions and actions have unfortunately made it impossible for Apollo to continue. Thank you so, so much for all the support over the years*. [Reddit Post]. Rerieved June 4, 2025, from https://www.reddit.com/r/apolloapp/comments/144f6xm/apollo_will_close_down_on_june_30th_reddits/

Jahanbakhsh, F., & Karger, D. R. (2024). *A browser extension for in-place signaling and assessment of misinformation* [Paper presentation]. In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, New York, NY, USA, May 1–21. Association for Computing Machinery. Retrieved October 27, 2025, from https://doi.org/10.1145/3613904.3642473

Jones, B. (2016). *Simplicity breeds success: The lasting impact of AOL Instant Messenger*. Retrieved November 28, 2025, from https://www.digitaltrends.com/computing/the-rise-and-fall-of-aol-instant-messenger/

Kalt, C. (2000). *Internet relay chat: Architecture* (Request for Comments No. 2810). Internet Engineering Task Force. Retrieved December 6, 2024, from https://datatracker.ietf.org/doc/rfc2810

Kantor, B., & Lapsley, P. (1986). *Network news transfer protocol* (Request for CommentsNo. 0977). USA: RFC Editor. Retrieved February 10, 2026, from https://www.rfc-editor.org/info/rfc0977

Koster, R. (2008). *A brief history of botting*. Retrieved December 10, 2025, from https://www.raphkoster.com/2008/03/25/a-brief-history-of-botting/

Latzko-Toth, G. (2010). Metaphors of synchrony: Emergence and differentiation of online chat devices. *Bulletin of Science, Technology & Society*, *30*(5), 362–374. https://doi.org/10.1177/0270467610380005

Lee, H. (2002). "No Artificial Death, Only Natural Death": The dynamics of centralization and decentralization of usenet newsgroups. *The Information Society*, *18*(5), 361–370. https://doi.org/10.1080/01972240290108177

Lewis, C. (1998). *Current spam thresholds and guidelines*. Retrieved June 5, 2025, from https://wiki.killfile.org/projects/usenet/faqs/spam/

McIlroy, M. D. (1971). *A research UNIX reader: Annotated excerpts from the programmer's manual, 1971-1986*. UNIX Programmer's Manual.

Merrill, J. B., & Tobin, A. (2019). *Facebook moves to block ad transparency tools—including ours*. Retrieved December 9, 2025, from https://www.propublica.org/article/facebook-blocks-ad-transparency-tools

Meta Platforms Inc. (2024). *Making messaging interoperability with third parties safe for users in Europe*. Retrieved April 9, 2024, from https://engineering.fb.com/2024/03/06/security/whatsapp-messenger-messaging-interoperability-eu/

Meta Platforms Inc. (2025). *Messaging Interoperability: WhatsApp enables third-party chats for users in Europe*. Retrieved November 24, 2025, from https://about.fb.com/news/2025/11/messaging-interoperability-whatsapp-enables-third-party-chats-for-users-in-europe/

Microsoft. (2008). *Microsoft Office Binary (doc, xls, ppt) File Formats*. Retrieved November 23, 2025, from https://web.archive.org/web/20080218212338/http://www.microsoft.com/interop/docs/officebinaryformats.mspx

Netscape Communications Corporation. (1998). *Netscape announces mozilla.org a dedicated team and web site supporting development of free client source code*. Retrieved November 28, 2025, from https://web.archive.org/web/19980706003741/http://www.netscape.com/newsref/pr/newsrelease577.html

Pfaffenberger, B. (1996). "If I Want It, It's OK": Usenet and the (outer) limits of free speech. *The Information Society*, *12*(4), 365–386. https://doi.org/10.1080/019722496129350

Rajendra-Nicolucci, C., Sugarman, M., & Zuckerman, E. (2023). *The three legged stool: A manifesto for a smaller, denser internet*. Retrieved December 6, 2024, from https://publicinfrastructure.org/wp-content/uploads/2023/03/The-Three-Legged-Stool-by-Chand-Rajendra-Nicolucci-Michael-Sugarman-and-Ethan-Zuckerman-iDPI-UMass-March-2023.pdf

Rauschmayer, A. (2014). Chapter 4. How JavaScript was created. In *Speaking Javascript*. O'Reilly Media, Inc. Retrieved November 28, 2025, from https://exploringjs.com/es5/ch04.html

Rentz, D. (2007). *The Microsoft compound document file format*. OpenOffice.org Source Project.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). *GroupLens: An open architecture for collaborative filtering of netnews* [Paper presentation]. In Proceedings of the 1994

ACM Conference on Computer Supported Cooperative work - CSCW '94, Chapel Hill, North Carolina. ACM Press, 175–186. Retrieved June 4, 2025, from http://portal.acm.org/citation.cfm?doid=192844.192905 https://doi.org/10.1145/192844.192905

Saint-Andre, P. (2004a). *Extensible Messaging and Presence Protocol (XMPP): Core* (Request for Comments No. 3920). Internet Engineering Task Force. Retrieved June 4, 2025, from https://datatracker.ietf.org/doc/rfc3920

Saint-Andre, P. (2004b). *Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence* (Request for Comments No. 3921). Internet Engineering Task Force. Retrieved June 4, 2025, from https://datatracker.ietf.org/doc/rfc3921

Schiller, T. (2021). *A brief history of browser extensibility*. Retrieved June 17, 2025, from https://medium.com/brick-by-brick/a-brief-history-of-browser-extensibility-bcfeb4181c9a

Smith, N. (2022). *Luna For Reddit*. Retrieved June 16, 2025, from https://www.nathantech.net/products/software/lunareddit.php

The European Parliament and the Council of the European Union. (2002). *Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Direcctive on privacy and electronic communictions)*(Vol. 201). Retrieved February 17, 2026, from https://eur-lex.europa.eu/eli/dir/2002/58/oj/eng

The European Parliament and the Council of the European Union. (2009). *Directive 2009/136/EC of the European Parliament and of the Council of 25 November 2009 amending Directive 2002/22/EC on universal service and users' rights relating to electronic communications networks and services, Directive 2002/58/EC concerning the processing of personal data and the protection of privacy in the electronic communications sector and Regulation (EC) No 2006/2004 on cooperation between national authorities responsible for the enforcement of consumer protection laws* (Vol. 337). Retrieved February 17, 2026, from https://eur-lex.europa.eu/eli/dir/2009/136/oj/eng (Usr lan: EN)

The 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)*. Retrieved January 9, 2026, from http://data.europa.eu/eli/reg/2022/1925/oj

The Matrix.org Foundation CIC. (2025). *Matrix specification*. Retrieved November 28, 2025, from https://spec.matrix.org/v1.16/

The Matrix.org Foundation CIC. (n.d.-a). *Bridges*. Retrieved June 4, 2025, from https://matrix.org/ecosystem/bridges/

The Matrix.org Foundation CIC. (n.d.-b). *Clients*. Retrieved June 4, 2025, from https://matrix.org/ecosystem/clients/

XMPP Standards Foundation. (n.d.). *History of XMPP*. Retrieved May 16, 2024, from https://xmpp.org/about/history/

Zhao, S., Wang, C., Kalra, A., Vaks, L., Borcea, C., & Chen, Y. (2017). *Ad blocking and counter-ad blocking ad blocking and counter-ad blocking: Analysis of online ad blocker usage* [Paper presentation]. In AMCIS 2017 Proceedings. Boston. https://aisel.aisnet.org/amcis2017/DataScience/Presentations/29

Zittrain, J. (2008a). After the stall. In *The future of the internet and how to stop it* (p. 70). Yale University Press.

Zittrain, J. (2008b). *The future of the internet and how to stop it*. Yale University Press.

Zuckerman, E., & Brickman, I. (2024). Improving social media with middleware. *The ANNALS of the American Academy of Political and Social Science*, *715*(1), 99–114. Retrieved December 16, 2025, from https://doi.org/10.1177/00027162251382700