Hard-coded censorship in Open Source Mastodon clients — How Free is Open Source?

Long paper

Juhani Naskali [0000-0002-7559-2595]

Information Systems Science,
Turku School of Economics, University of Turku
Turku, Finland
juhani.naskali@utu.fi

Abstract. This article analyses hard-coded domain blocking in open source software, using the GPL3-licensed Mastodon client Tusky as a case example. First, the question of whether such action is censorship is analysed. Second, the licensing compliance of such action is examined using the applicable open-source software and distribution licenses. Domain blocking is found to be censorship in the literal definition of the word, as well as possibly against some the used Google distribution licenses — though some ambiguity remains, which calls for clarifications in the agreement terms. GPL allows for functionalities that limit the use of the software, as long as end-users are free to edit the source code and use a version of the application without such limitations. Such software is still open source, but no longer free (as in freedom). A multi-disciplinary ethical examination of domain blocking will be needed to ascertain whether such censorship is ethical, as all censorship is not necessarily wrong.

Keywords: open source, FOSS, censorship, domain blocking, licensing terms

1 Introduction

New technologies constantly create new challenges. Old laws and policies cannot always predict future possibilities, and sometimes need to be re-examined. Open source software is a licensing method to freely distribute software code, but also an ideology of openness and inclusiveness, especially when it comes to FOSS (Free and Open-source software). But how does openness and inclusiveness coexist with intolerance and possibly hateful content?

Mastodon is a free and open-source social network, sometimes also called a microblogging service. Notable features of Mastodon include interoperability and federation, which allow everyone to run their own instance of Mastodon or another standard complying server software, and have their users' messages federated to other instances using the ActivityPub standard. (Rochko, 2018; TootSuite, 2019) This allows people

using one site or instance to communicate with users on other instances, similarly to how email works. One characteristic of a federated system is that it is decentralized, so there is no single entity that has the ability to censor all users or posts. Each Mastodon instance has its own rules for membership and moderation, and Mastodon includes tools for individual users to block messages from specific users or instances.

While it is possible to use Mastodon with a browser, it can be more convenient to use a native client application. There are many Mastodon clients for desktops and mobile devices, which allow you to communicate using your Mastodon account, no matter which instance you are using — or at least that used to be the case.

Gab (https://gab.com) is a microblogging instance based on Mastodon, which claims that it ”champions free speech, individual liberty and the free flow of information online”, but is considered by many to contain extreme hate speech. Gab has caused uproar in the media, and has subsequently been blocked by Paypal (McKay, 2019). Their mobile applications have been removed from Google Play Store and Apple Store (Lee, 2017).

Many Mastodon client programs have now implemented hardcoded blocks for Gab users. This paper examines the legality and ethicality of open source client programs hard-coding blocklists in their applications, using the Android Mastodon client Tusky and its GPL-3.0 (GNU General Public License v3.0) licensing terms as a case example.

2 Case: Rickrolling Gab users in Mastodon client Tusky

2.1 Rickrolling instead of logging in

On 17th of June 2019, a change was made to the Mastodon social media client Tusky that prevents some users from logging in, and instead redirects unsuspecting users to a famous 1987 music video of Rick Astley’s song ”Never Gonna Give You Up” in a common internet gag familiarly coined as ”rickrolling”. The codechange is relatively simple. It checks if the user’s domain is ’gab.com’ or ’gab.io’ (or a subdomain thereof), and opens a browser view of the specified youtube url based on this check. (Tusky, 2019a) The change renders the app unusable with Gab accounts, as they are unable to log in.

The officially distributed versions of Tusky include this functionality and can be downloaded from F-droid, Google Play or Amazon Appstore. Removing the block requires changing the code and compiling it yourself, which is outside the expertise of most smartphone users.

2.2 Functionality announcement and discussion

Tusky’s block-implementing merge commit was made with the comment ”Rick roll instead of logging in on selected domains. This is not censorship, but rather a choice by this house who will facilitate our services to.” (Tusky, 2019a) There are no comments on the code changing merge commit, but the pull request introducing the code change gathered 167 comments before it was locked down as off topic by the developers (Tusky, 2019b).

The Tusky pull request mentions it is a copy of a similar earlier implementation in Sengi app, where a commit titled ”added a little check” created the rickroll effect for the same group of users (NicolasConstant, 2019). The change in Sengi was announced on Mastodon in a humorous toot (a Mastodon post, similar to a tweet on Twitter) containing a video clip and the text ”Here is what will happen if someone tries to log-in with a #gab account in #sengi :smirk:” (Sengi, 2019).

While Sengi's change made it through without much notice, Tusky's pull request generated a lot of heated discussion. Some people commented in favor of the change while others were against it, thought it might be illegal — or agreed in principle but thought that rickrolling was not the most productive way of blocking people. Fascism and Nazis were mentioned. The moderation team ended up locking the comment thread citing the amount of moderation work required. (Tusky, 2019b)

F-Droid is a repository for FOSS (free and open-source software) on Android platform. They issued a statement (F-droid, 2019) after a lot of user discussion about whether they should block Mastodon clients based on whether they implement blocking or not. The discussion was later taken down, but can still be accessed through the Internet Archive at https://web.archive.org/web/20190711065044/https://forum.f-droid.org/t/tusky-is-nonfree/6448.

All in all, the topic has generated quite a wide array of discussion and elicited strong emotions in people. There is no clear consensus on whether such domain blocking is in accordance to GPL licensing, whether it is considered censorship or whether it is ethical.

2.3 Other Mastodon clients

Gab is also censored on other Mastodon clients: Amaroq, Mastalab, Mastodome have all implemented similar blocking features (Mastodome, 2019; Fedilab, 2019b; Amaroq, 2019) and Toot! on iOS also blocks Gab users according to their Mastodon announcement[1].

> *”- I didn't receive a reply from Google policy team about a potential ban of the application.*
> *- Without the previous risk, I clearly think that's not my role. I can't hard-code instance blocks especially when every tools are [sic.] here for that.*
> *- If you want a strong block, it's in the hands of social network developers or your admins.” (Fedilab, 2019a)*

Mastodon does include tools for users to block other users or instances from showing for themselves, and there are tools for instance administrators to do the same for all their users. Nevertheless, the percentage of Mastodon apps implementing hard-coded domain block is quite high, which means that while it's not true that blocked users do not have access to any Mastodon clients — and indeed even if this was the case, they could still use a browser to use their chosen service, even without coding skills — their options are noticeably limited.

[1] https://mastodon.social/@tootapp/102185365923885685

After domain blocking was implemented, an uncensored version of Tusky was released on Google Play[2], making it possible for Gab users to keep using Tusky. This unofficial version was later blocked by Google. Even if modified versions would still be available, using them would require users to trust the distributor of the unofficial version.

2.4 Reasons for using Tusky as a case study

Censorship in sourcecode is highly interesting to begin with, and this is one of the first recorded cases of hard-coded censorship based on the user's chosen platform, instance or service provider. It has also been traditional for client programs to be provider-agnostic, working on any and all providers of the supported protocols, and this type of service-specific blocking is new, especially in open source software. It is technically analogous to Outlook email program not working for Gmail users (though such a hypothetical case would be ideologically very different).

This type of service restriction is not directly considered in current licensing and policy texts. It is possible that the emergence of this new type of restriction on users necessitates some reviews in open source licensing terms and/or developer policies. One of the motivations with GPL-3.0 version was to restrict the use of DRM (Digital Rights Management) and patents to limit the rights of end-users to change software:

> *"It [GPLv3] doesn't forbid DRM, or any kind of feature. It places no limits on the substantive functionality you can add to a program, or remove from it. Rather, it makes sure that you are just as free to remove nasty features as the distributor of your copy was to add them.*
> *[...]*
> *The explicit patent license of GPLv3 makes sure companies that use the GPL to give users the four freedoms cannot turn around and use their patents to tell some users, 'That doesn't include you.'" (Stallman, 2014)*

3 Is hard-coded domain blocking censorship?

3.1 Definition of censorship

While the Tusky commit message claims this is not censorship (Tusky, 2019a), it clearly fits the dictionary definition in Encyclopedia Britannica:

> *"Censorship, the changing or the suppression or prohibition of speech or writing that is deemed subversive of the common good. It occurs in all manifestations of authority to some degree, but in modern times it has been of special importance in its relation to government and the rule of law."*

The code change does suppress writing by Gab users, so it fits the definition of censorship. The developers and distributors of Tusky have authority over the app and

[2]https://play.google.com/store/apps/details?id=codes.lin.tuskyuncensored

their userbase, though their authority is not as direct as in case of governments, where censorship is backed by law.

Merriam-Webster's definition of the verb censor is even more terse: *"to examine in order to suppress or delete anything considered objectionable censor the news also : to suppress or delete as objectionable"* and also fits domain blocking.

3.2 Hard and soft censorship

One reply to Tusky's Pull request (Tusky, 2019b) by GitHub user twisterghost (Michael Barrett) brings up a compelling point: "As stated before, tusky is a convenience layer and open source. The official app and the maintainers are humans, and they have every right to take the app in whatever direction they see is best. If people absolutely must, they can fork and remove this limitation, and distribute that. The license explicitly allows it."

The point about Tusky being a "convenience layer" is a valid one — the censored users can still speak freely using other means. They are only blocked from using specific software. As such it cannot be considered a case of hard censorship, where the content of the expressed opinions is deleted, but it can be likened to soft censorship, where opinions are suppressed by withholding payments or applying other kinds of pressure. More on this in 3.4.

3.3 Transparency, chilling effect and other censorship considerations

There are some common considerations, when examining cases of soft censorship. One of these is the chilling effect of unclearly defined or untransparent censorship. If there are rules in place that punish (e.g. through monetary losses) for publishing some content, but the limits of what can and cannot be said are unclear, this acts as deterrence — a chilling effect — against any content that could conceiveably be considered controversial, even if such content would not clearly break any explicit rules (Schauer, 1978).

Open source is, by nature, transparent. All censored domains can be clearly seen in the freely posted source code (Tusky, 2019b), so knowledge of what is censored is public, though perhaps not easily found. As such, there is no opaqueness in the actual censorship process, which could contribute to a chilling effect.

There still might be a slight chilling effect caused by the unclear reason given for this act of censorship, as it seems that the change was accepted only because the creators of Tusky deemed Gab to be disagreeable to them, and by their definition, harmful. They might be correct in their assessment, but the users of Tusky are subjected to subjective decisions on what to censor and what not to censor. Still, it is unclear how this could contribute to a chilling effect, as users' activities are not censored directly.

The chilling effect should also be considered in the wider picture of such blocks also being implemented in other ways, such as Mastodon instances blocking other instances. Prima facie, it seems like the psychological effects of blocking instances or platforms instead of user activities would be considerably different. This question, however, is outside the scope of this paper.

3.4 Blocking users based on their service provider is censorship

While hardcoded domain blocking in a particular software application only prevents users from accessing their service via that specific application, it still suppresses a specific means of expression. This fits the dictionary definition of censorship, even though the action is very different from hard govermental censorship, where published content is removed. Historically, only censorship of news outlets was considered censorship because little else was pertinent to the spread of expression. In modern times, it is possible to censor the spread of information without actually removing content, by for example removing search results or preventing access with blocklists in the readers' internet connection. Such acts are censorship, even if they do not directly remove published material.

Imagine a keyboard manufacturer whose keyboards do not work for writing anything that goes against the company's political views. Avoiding one such keyboard brand to express yourself would be irritating. Avoiding ten would get difficult. If all keyboard manufacturers (hardware and software) join the blocking effort, we get something close to total censorship. While such imaginary censorship keyboards are far from practical, there are some parallels to blocking users inside applications. While domain blocking is not complete censorship of certain viewpoints, especially when not widely coordinated, it would be disengenious to say it is not censorship at all.

Another, perhaps more fruitful, perspective is to note that software utilizing domain blocking is no longer free in the sense that it doesn't allow its users to use the software freely, but instead imposes limits on how the software is to be used. Sidestepping the definition of censorship, it is clear that the software's use is being limited. Such limits are not necessarily wrong in the moral sense. Developers own rights to their code, and open-source licenses such as GPL do not limit them in what functionalities they can include in the application.

Open-source software that utilizes domain blocking is a good example of software that is open-source but non-free. The first (freedom 0) essential freedom of free software is "The freedom to run the program as you wish, for any purpose" (GNU Project, 2001), and this freedom is not granted to domain-blocked users. While domain blocking is not hard censorship, it does fall into the continuum of soft censorship.

4 Legality of censorship

4.1 Considerations

Programmers have copyright to their creations and are quite free to dictate how their code and applications can be used and by whom. These rights can possibly be subject to preceding rights by others — rights that create more compelling duties to respect other people's freedoms. The creators can also willfully give away parts of these rights to others with contracts and agreements.

In the case of open source, the creators of software code enter into a licensing agreement with others, guaranteeing their right for using and modifying the code freely. Tusky is licenced under GPL-3, so the GPL license will be examined in more detail, though many of the findings might be applicable to other similar licenses.

Further considerations are any agreements entered in the distribution of applications. For example in the case of Tusky, app publishers need to agree to the Google Play Developer Distribution Agreement and Developer Program Policies, which require them to adhere to certain rules for distribution through Google Play Store (Google, 2019a,b). These agreements differ from platform to platform, and might restrict app developers rights in different ways when it comes to the distribution of compiled applications. Notably these considerations no not relate to the open source code but the compiled executable application, which users download and run directly.

4.2 Possible violations of GPL-3

Definition of Free

The definition of Free Software has been discussed since the 80s, with GNU's 1st bulletin being one of the first records of what is considered free software: "When we [the Free Software Foundation] speak of free software, we are referring to freedom, not price." (GNU, 1986, p.8) This definition of free software focuses on user freedom, but mainly discusses the freedom to share, read and modify code.

Open Source Initiative states that open source by definition should comply with "No Discrimination Against Persons or Groups" and "No Discrimination Against Fields of Endeavor" (Initiative, 2019). Users of a specific Mastodon instance constitute a distinct group, especially if instance selection is based on an ideology or other identifiable characteristic, which means that domain blocking doesn't comply with this criteria.

Interestingly, this point of contention was — at least partly — already considered around 1990 by the GNU Project, which added a "freedom 0" to their text, which precedes the freedoms relating to study, redistribution and modifying the code: "The freedom to run the program as you wish, for any purpose". (GNU Project, 2001) Stallman (2013) later expanded on the reasons why programs must not limit the freedom to run them by licensing, explicitly stating that distributions shouldn't restrict how you use the software.

> *"It is worse than ineffective; it is wrong too, because software developers should not exercise such power over what users do. Imagine selling pens with conditions about what you can write with them; that would be noisome, and we should not stand for it. Likewise for general software. If you make something that is generally useful, like a pen, people will use it to write all sorts of things, even horrible things such as orders to torture a dissident; but you must not have the power to control people's activities through their pens. It is the same for a text editor, compiler or kernel." Stallman (2013)*

Notably, Stallman was writing specifically about license restrictions. If the ideal of freedom is user choice, it is difficult to see how hard-coded limits would fit this ideal. But when it comes to open source the ideal is to secure users freedom in relation to the source code, and not the distributed program(s). While not related to the license, it is worth noting that such distinction only secures freedoms for those people who have the necessary skills to alter code and compile the programs, discriminating against those

who do not have the means to remove restrictions for themselves or cannot have others do it or them.

While restricting the usage of programs due to ideology is arguably against the spirit of free software, akin to Stallman's above example of dictating what can be written using a pen, it is not against the licensing terms of GPL-3.0. Stallman's explicit acceptance of DRM, as long as it can be removed, speaks to this directly (Stallman, 2014). Such software code is still open-source, but the distributed software application is no longer free. This distinction speaks volumes to the need for terms such as FOSS (Free and Open Source Software) that makes the distinction between free as in freedom and "merely" open source.

Permission to run the program

Line 158 of the GPL-3 license states "This License explicitly affirms your unlimited permission to run the unmodified Program." The license goes on to prohibit different ways of limiting the use of software, including use of DRM (digital rights management), that cannot be removed, as well as withholding installation information.

It has been argued that allowing the program to run in this manner is enough for freedom 0 (Slep, 2019), as only the functionality of the application is limited — not its running, as blocking is done after the program starts. While domain blocking technically allows the program to be run, it only does so similarly to a DRM system that runs only to disallow the usage of said program.

Some DRM systems require a login after the application runs, and block access if the login is not licensed to use the program. It would be disengenious to claim that in either case the program runs for DRM-blocked users, when it fulfills none of its intended functions for non-licensed users. Still, even this type of DRM is explicitly allowed in GPL-3 licensed applications. As long as users are free to change the source code and run their own versions of the programs without such restrictions, Freedom 0 is retained to those who can compile their own programs.

While domain blocking is akin to DRM in that it blocks the program from working based on the used media (Mastodon instance or DVD, respectively), this is not against the GPL.

Denial of Access

GPL-3 line 333 states: "Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network." Insofar as domain blocking can be construed as denying access to a network (such "deny rules" usually affect network traffic and not user input that is expected to translate to network traffic), domain blocking could be in violation of GPL.

However, such interpretation of network access would be shaky at best, and not the intended meaning of this part of the license. The line is noted here only for completeness sake.

4.3 Violations of Google's policies

Interruption of service

It could be tenuously argued that censoring a Mastodon instance constitutes as disruption of service. The developer agreement states: "4.9 You will not engage in any activity [...] that interferes with, disrupts, damages, or accesses in an unauthorized manner the devices, servers, networks, or other properties or services of any third party" (Google, 2019b).

However, the word disruption generally refers to larger disruptions to a network or a system, just as in the case of GPL, and not to specific blocking of user's intended services. As such, domain blocking does not seem like a violation of this clause.

Product takedowns

Google's Agreement prohibits product takedowns from users that have previously purchased or downloaded the products:

> *8.1 You may remove Your Products from future distribution via Google Play at any time, but You agree to comply with this Agreement and the Payment Processor's Payment Account terms of service for any Products distributed via Google Play prior to removal including, but not limited to, refund requirements. Removing Your Products from future distribution via Google Play does not (a) affect the rights of users who have previously purchased or downloaded Your Products; (b) remove Your Products from Devices or from any part of Google Play where previously purchased or downloaded applications are stored on behalf of users; or (c) change Your obligation to deliver or support Products or services that have been previously purchased or downloaded by users.* (Google, 2019b, 8.1c)

Gab users that used Tusky before domain blocking was implemented suddenly found themselves rickrolled instead of getting the service they were using. As modern phones often automatically install updates, users might have no way to prevent loss of service due to updates. It could be argued that blocking gab.com users that had previously been able to use the app breaks this clause of the agreement.

From another point of view, the wording of the clause seems to only affect removals of complete applications and not removal of functionality. However, if one considers functionality removals OK under the clause, this would allow removal of any program by replacing it with a placeholder application with no functionality — at least when done in a targeted manner (e.g. using domain blocking).

Without clarification from Google, the clause remains partially unclear. Nevertheless, there seems to be grounds to argue that removing functionality from users who previously downloaded the application is a violation of this policy.

Deceptive behaviour

Github user Vaasref commented on Tusk's pull request (Tusky, 2019b) that he needs to trust apps installed on their phone, and suggested that the app doing something against

the will of the user is in violation of Google Play Developer Policy, which states that Google doesn't allow "apps that attempt to deceive users or enable dishonest behavior".

Google's Developer Policy states that *"Apps must provide accurate disclosure of their functionality and should perform as reasonably expected by the user."* Users can reasonably expect a service client to log in and function as a service client, so domain censoring in this way constitutes a policy violation. Also, not disclosing such censorship functionalities (e.g. in the app description) constitutes a violation.

Tusky's domain block could constitute as deceptive behaviour if the user doesn't get any information on why, instead of logging in, they are now watching a video of Rick Astley from 1987. Clear explanation of the blocking functionality and a descriptive error messages would alleviate this infringement.

Minimum functionality

Google's Developer Policy also requires the following:

> *"At a minimum, apps should provide users with a basic degree of functionality and a respectful user experience. Apps that crash, exhibit other behavior that is not consistent with a functional user experience, or that serve only to spam users or Google Play are not apps that expand the catalog in a meaningful way."* (Google, 2019b)

Blocking logging in to a service on an application whose only function is to work as a client to use said services clearly violates this policy. Furthermore, doing so by rickrolling isn't respectful, and constitutes another clear violation of this policy. Even with a respectful explanation of the intended blocking functionality and a descriptive error message during the blocking, it is difficult to consider not logging in as providing "a basic degree of functionality" "consistent with a functional user experience".

5 Conclusion

Domain blocking through rickrolling doesn't seem to be against GPL, but does arguably violate Google's Developer Program Policies, as it blocks the user from the application's minimum functionality. Furthermore, removing access from previous users seems to also violate Google Play Developer Distribution Agreement. Nevertheless, the clauses are unclear when it comes to blocking functionality in this way, and require clarifications.

GPL-3.0 allows for functionalities that limit user rights, such as DRM, as long as such limitations can be removed from the source code by the user and the newly created program can be used without limitations. Thus domain blocking, as another way to limit usage rights digitally, is allowed by the licensing terms, and programmers have not signed away their right to decide how the application they distribute will be used (though end-users are free to change the source code and overrule any imposed limits). However, distributed applications with such limitations can no longer be considered free, as they limit the freedom of users to use the program as they wish.

This does create some inequality between those who have the technological knowledge necessary to bypass these limitations and those who do not. However, such questions about the morality and unintended consequences of domain blocking would require a multi-disclipinary ethical examination of the case, which is outside the scope of this paper.

Even though domain blocking doesn't seem to violate GPL, blocking all application functionality by rickrolling them based on users' selected communication group or ideology does seem at odds with the general ideology of free software, and distributing such censorship functionality in software should be done carefully, with transparency and respect. To do otherwise seems to be against Google's Developer Distribution Agreement, and there might be similar clauses in other platforms' distribution agreements. Implementing domain blocking under GPL creates software that is open-source and non-free, which is an interesting space to inhabit, especially when it comes to questions of free speech and censorship. The ethical considerations of such cases clearly require more research.

References

Amaroq (2019). Hardcode ban of gab.anything · reticentjohn/amaroq@92afdbd.
Retrieved from https://github.com/ReticentJohn/Amaroq/commit/92afdbd3309176cd927364090d1ab0c058cc2f12

F-droid (2019). Public statement on neutrality of free software.
Retrieved from https://f-droid.org/en/2019/07/16/statement.html

Fedilab (2019a). Fedilab announcement.
Retrieved from https://framapiaf.org/@fedilab/102299778188330713

Fedilab (2019b). Fix crash + add gab.ai block · grufwub/fedilab@8b50ce0.
Retrieved from https://github.com/grufwub/fedilab/commit/8b50ce0b2a00eb6bb12641cb4bdd0f1a1dc63035

GNU (1986). Gnu's bulletin volume 1 no.1.
Retrieved from https://www.gnu.org/bulletins/bull1.txt

GNU Project (2001). What is free software?
Retrieved from https://www.gnu.org/philosophy/free-sw.html

Google (2019a). Developer policy center.
Retrieved from https://play.google.com/about/developer-content-policy/

Google (2019b). Google play developer distribution agreement.
Retrieved from https://play.google.com/intl/ALL{_}us/about/developer-distribution-agreement.html

Initiative, O. S. (2019). The open source definition.
Retrieved from https://opensource.org/osd

Lee, T. B. (2017). Google explains why it banned the app for gab, a right-wing twitter rival.
Retrieved from https://arstechnica.com/tech-policy/2017/08/gab-the-right-wing-twitter-rival-just-got-its-app-banned-by-google

Mastodome (2019). Updated project status and added block for gab · treacherousnexus/mastodome-legacy@1a87efd.
Retrieved from https://github.com/TreacherousNexus/mastodome-legacy/commit/1a87efdaf3b610ce5438bae612079ab309e950ce

McKay, T. (2019). Paypal bans far-right social network gab after anti-semitic user kills at least 11 at synagogue.
Retrieved from https://gizmodo.com/paypal-bans-far-right-social-network-gab-after-anti-sem-

NicolasConstant (2019). Commit "added a little check" nicolasconstant/sengi@cf83f73.
Retrieved from https://github.com/NicolasConstant/sengi/commit/cf83f73

Rochko, E. (2018). Why activitypub is the future - official mastodon blog.
Retrieved from https://blog.joinmastodon.org/2018/06/why-activitypub-is-the-future/

Schauer, F. (1978). Fear, risk and the first amendment: Unraveling the chilling effect. *BUL rev.*, *58*, 685.
Retrieved from https://scholarship.law.wm.edu/cgi/viewcontent.cgi?article=2010&context=facpubs

Sengi (2019). Sengi: "here is what will happen if someone tries to log-in" - mastodon.
Retrieved from https://mastodon.social/@sengi{_}app/102194358508642906

Slep, C. (2019). On tusky rickrolling.
Retrieved from https://cjslep.com/c/blog/on-tusky-rickrolling

Stallman, R. (2013). Why programs must not limit the freedom to run them.
Retrieved from https://www.gnu.org/philosophy/programs-must-not-limit-freedom-to-run.html

Stallman, R. (2014). Why upgrade to gplv3.
Retrieved from https://www.gnu.org/licenses/rms-why-gplv3.en.html

TootSuite (2019). Mastodon on github.
Retrieved from https://github.com/tootsuite/mastodon

Tusky (2019a). Merge pull request #1303 from mlc/rick_roll_domains · tuskyapp/tusky@5d04a7c.
Retrieved from https://github.com/tuskyapp/Tusky/commit/5d04a7c

Tusky (2019b). Tusky pull request #1303 · tuskyapp/tusky.
Retrieved from https://github.com/tuskyapp/Tusky/pull/1303/