Commentary

Nuances and Challenges of Moderating a Code Collaboration Platform

Margaret Tucker, Rose Coogan, and Will Pace

1 Introduction

"Online platform" is an exceptionally broad term that has been used to describe social media networks, marketplaces, search engines, communication services, and many more internet services (West 2019). It comes as no surprise that the majority of studies focusing on the content moderation practices of online platforms has focused on the few "very large online platforms," to borrow a term from the European Union's Digital Services Act,[1] with the highest user counts and general-purpose uses. It is certainly important to understand how the largest platforms function, especially when they have massive influence over modern speech and society, but platforms with specialized purposes and niche user groups have significant lessons for the field of Trust and Safety and platform studies writ large. This commentary offers GitHub as an example of an atypical platform whose purpose and user group has driven its content moderation practices. It will explore what makes a code collaboration platform different from other platforms, how addressing code as content influences GitHub's content moderation approach, and how this approach is driven by the norms and needs of the software developer community.

1.1 What is GitHub

Founded in 2008, GitHub is a developer platform that enables users to host, share, and collaborate on software code. Since its inception, GitHub has grown to become the largest code collaboration platform, with over 100 million users working together on code throughout the world. GitHub is a significant steward of the world's open source[2] software, which is especially important because open source software is an integral component of the world's digital infrastructure: in a survey of 1,067 commercial codebases across 17 industries, 96% contained open source components (Synopsys 2024). GitHub's role as a home for the open source software community has driven substantial investments in securing open source at scale. In 2021, GitHub launched GitHub Copilot, an AI-driven pair programmer that provides real-time code suggestions and chat assistance in natural

1. See DSA: Very large online platforms and search engines
2. See Open Source Initiative open source definition.

doi:10.54501/jots.v2i4.213

language recommendations as developers work. Developers are using GitHub for a range of AI development, which introduces new challenges for platform governance.

1.2 Categorizing code collaboration platforms

While GitHub and other code collaboration platforms share characteristics with platforms oriented toward content creation, productivity, education, professional purposes, and the broader open source community, they best fit under the category of a user-generated content service or productivity platform. That said, GitHub does share some similarities with social media platforms: users can create a GitHub profile, which developers often use as a portfolio. Code collaboration is inherently social, and while GitHub does not have private messaging, users can interact with each other in comments; follow other users, projects, and repositories; and receive update notifications. Users can search for relevant projects, electing to sort by the best match according to the search query, the most or fewest stars, forks, or time of last update. GitHub has a dashboard with a customizable feed that gives users control over what updates they see, including users and projects that may be relevant to their interests.[3] Yet there is limited capacity to "go viral" on GitHub; when significant incidents on GitHub have led to virality and media attention, they are typically shared and amplified on a general-purpose social media platform rather than GitHub itself. GitHub also has similarities to a website hosting service: GitHub Pages[4] a static hosting service that allows users to host a website about themselves, their organizations, or their projects using files from their repository. However, GitHub Pages is intended to be used solely for sites showcasing projects being developed on GitHub, and the use of Pages as a free web hosting service for commercial services is prohibited. Code collaboration platforms' distinctiveness necessitates a nuanced approach to content moderation.

1.3 Revenue models of platforms differentiate incentives

The incentives and structure of platforms are driven by their revenue models. GitHub largely generates revenue through its freemium model, which offers free repository hosting for public projects and paid hosting for private repositories and additional features for enterprises and individuals, rather than advertising or user data sales, which sets it apart from general-purpose social media platforms (Bounegru 2023). This lack of targeted advertising shifts GitHub's incentives toward enhancing its utility for software developers, positioning it to compete effectively with other code collaboration platforms.

These different revenue models influence the type of content appearing on each platform. Traditional social media platforms rely heavily on advertising revenue, which is driven by user engagement. Engagement-based algorithmic feeds often amplify divisive content (Milli et al. 2024). In contrast, code collaboration platforms like GitHub focus

3. See GitHub Blog Updates to your GitHub feed.
4. See About GitHub Pages.

on professional, business-related content. They avoid advertising and tracking, instead generating revenue through long-term investments in projects and businesses. While they benefit from increased user numbers and usage, the specific content is less relevant. GitHub, for example, charges per user or use of compute,[5] regardless of the content, functioning similarly to infrastructure services like broadband access or web hosting, where revenue depends on usage scale rather than content type.

2 Code is a different type of content, presenting specific moderation concerns

Another significant distinction of GitHub is the nature of the content it hosts: software code. Unlike photos, videos, or audio, code is text that instructs a computer to perform specific functions. While code can be expressive ("code is speech"), its primary purpose remains functional, which creates distinct considerations for platform moderation (Dame-Boyle 2015). Common programming solutions often have limited variations, leading to widespread independent duplication that creates unique considerations for copyright. The functional aspect of code also necessitates caution whenever removing or disabling access to code that may be in use. Software code, particularly open source software, functions as digital infrastructure, underpinning a vast array of applications and services, including critical infrastructure, used globally (Scott et al. 2017). As such, moderating a code collaboration platform like GitHub demands careful consideration to ensure that essential code remains accessible. Moreover, a code collaboration platform must prioritize mitigating the potential harms of malicious code, such as malware, due to the direct and severe impact malicious code can have on software systems and users.

2.1 Copyright concerns

Software code has unique copyright concerns. Its functional aspect leads to inevitable duplication, particularly within open-source software development, where sharing is encouraged by copyright licenses. Code shared on developer platforms like GitHub are typically licensed under terms that allow and encourage sharing and remixing. Often, copyright disputes arising on sites like GitHub are not about using another developer's code but about meeting specific requirements, such as attribution, under the terms of the relevant licenses. Many licensing disputes can be resolved between developers without GitHub's intervention. This norm of sharing and reuse exacerbates the challenge of creating effective filtering technologies to detect copyright infringement. Partial code matches often include innocuous, reused code, making it difficult to accurately identify infringing uploads without generating massive false positives. Filtering technologies can thus disrupt interdependent code ecosystems with shared functionalities.

5. See GitHub's plans and features pricing.

2.1.1 Case Study 1: EU Copyright Directive

When the EU Copyright Directive was first introduced in 2018, it included provisions that would have required platforms to implement upload filters to prevent copyright infringement (Reynolds 2019). Before introducing this new law, the EU's protections for platforms that host user-generated content were similar to those in the US Digital Millennium Copyright Act (DMCA),([6])which has safe harbor provisions for platforms, shielding them from liability for user's copyright infringement in exchange for complying with the notice and takedown process and other requirements. Safe harbor provisions like the DMCA allow platforms to host user-generated content without having to proactively scan and review all content for potential infringement(Stoltz 2019). The new proposed requirement raised significant concern within the software developer community about the high likelihood of false positives that would inevitably result from implementing upload filters for code. Moreover, there was not a strong case for mandating filters for code collaboration platforms at all: the "value gap" argument that platforms use advertising to monetize potentially infringing user-uploaded media without compensating rightsholders does not hold for platforms like GitHub.

In response, GitHub led a collaborative advocacy effort[7] to gain an exemption for code collaboration.[8] As a result, the final version included an exclusion for "open source software development and sharing platforms," mitigating the risk of false positives and ensuring continued innovation in software development. The importance of excluding software development platforms from mandatory content filtering requirements illustrates the complexities involved in detecting copyright infringement in software, the vast differences in incentives for code collaboration platforms versus other content hosting platforms, and the unique needs of the developer community that policymakers must consider.

2.1.2 Case Study 2: Developers and the DMCA

Legal scholars have noted that the DMCA contributes to chilling effects on software development, leading to excessive caution and self-censorship among developers (Penney 2019). Its anti-circumvention provisions hinder legitimate activities like security research, reverse engineering, and interoperability efforts due to fear of legal liability. Additionally, the DMCA's notice-and-takedown system can be exploited through false or abusive claims, resulting in the removal of non-infringing content and discouraging developers from sharing innovative work. These broad and often vague provisions create fear, uncertainty, and doubt, leading to over-censorship and legal risks that stifle creativity and collaboration. The liability threat for content hosting platforms like GitHub can lead them to impose content filtering systems that can inadvertently remove non-infringing content (Schaffner et al. 2024).

6. See U.S. Copyright Office fact sheet for 17 U.S.C. § 512.
7. See GitHub Blog The EU Copyright Directive: what happens from here.
8. See GitHub Blog How developers can defend open source from the EU copyright proposal.

GitHub's DMCA policy[9] was designed to suit the code collaboration environment. When only certain content within a repository is identified as alleged copyright infringement, GitHub provides users with the opportunity to make changes to the specific files identified as infringing before disabling the repository as a whole, to preserve the availability of code as much as possible. Downstream fork networks of repositories are not automatically disabled without confirming they also contain the allegedly infringing content, and developers are provided with a path to dispute infringement claims. GitHub maintains transparency by posting redacted copies of legal notices[10] and conducting extensive reviews of complex circumvention claims[11] to protect the collaborative environment essential for software development.

2.2 Network effects of takedowns

Code takedowns have cascading effects that can harm developer ecosystems. Code on GitHub may be in use by millions of computers around the world, and a wrongful takedown can have enormous consequences to the developer ecosystem. In modern software development, programmers write code that "depends" on other tested, proven, and widely accessible software—usually open source software—written by third parties. All types of software, from phone apps to enterprise software run by corporations and governments, rely on these "dependencies." When even a single dependency is removed from a software collaboration platform like GitHub in response to a takedown request, its removal can break the software of an exponential number of other programs that depend on that code.

2.3 Dual-use software

Software code often has dual-use applications, serving both beneficial and harmful purposes depending on its implementation. This duality is particularly relevant in security research, where code designed to identify vulnerabilities can also be exploited to create malware or execute cyberattacks. Similarly, techniques developed to bypass copyright protection for legitimate purposes, such as interoperability, accessibility, or security testing, can be misused for piracy or unauthorized access.

GitHub recognizes that the functional nature of code allows for varied applications and evaluates whether code may have dual uses or is designed solely for abusive purposes. There is an expectation that security research will be clearly labeled to prevent users from unintentionally downloading malware. By allowing dual-use software to be shared openly, GitHub supports the use of its beneficial applications as well as efforts to counteract harmful uses. This stance is informed by the values of the software developer community, whose oversight ensures transparent and responsible use.

9. See GitHub DMCA Takedown Policy.
10. See GitHub DMCA takedown repository.
11. See GitHub Blog Standing up for developers: youtube-dl is back.

2.3.1 Case Study 3: Update to malware policy

GitHub has refined its approach to dual-use security research over time. In 2021, GitHub initiated a 30-day notice-and-comment period to gather community feedback on proposed changes to its policies on exploits, malware, and vulnerability research.[12] The goal was to enable, welcome, and encourage good-faith security research. Driven by this feedback, GitHub clarified that it explicitly permits the posting of proof-of-concept exploits and dual-use security technologies essential for legitimate practices like penetration testing. However, GitHub disallows the use of its platform for active attacks that cause technical harm, such as malware campaigns or denial-of-service attacks. The policy update aimed to eliminate ambiguity and foster an environment where beneficial security research can thrive while preventing abuse. Additionally, GitHub introduced an appeals process to handle disputes over content removal, promoting transparency and collaboration between security researchers and the platform.

2.4 Moderating code requires different considerations

The specificities of code as content, along with the interdependent nature of code collaboration, means that moderating a platform like GitHub requires technical understanding and careful consideration of context, dual-use applications, and network effects of takedowns. It also means that the proportion and categories of Terms of Service[13] violations GitHub encounters are distinct from those found on general-purpose media platforms; along with general abuse, GitHub encounters malware, copyright and trademark infringement, cryptocurrency abuse, and exposed data and personal information. GitHub has developed its content moderation approach to suit the nuances of code collaboration. While it has integrated some automated flagging for spam, for other Terms of Service violations, the nuances of code as content and the broader impacts of takedowns necessitate the use of human reviewers. GitHub aims to have a developer-first approach to content moderation, minimizing the disruption to collaboration on legitimate content while addressing abuse and other violations.[14]

GitHub also encounters issues when its features are used for activities unrelated to code collaboration. For example, GitHub Pages is solely intended[15] for sites about GitHub users, organizations, and projects being developed on GitHub, and users of Pages must comply with the GitHub Terms of Service. Pages creates additional moderation challenges for GitHub Trust and Safety because of the wide range of off-topic, prohibited uses outside of Terms of Service violations; GitHub must make continual efforts to keep the use of Pages relevant to the developer community.

12. See GitHub Blog Updates to our policies regarding exploits, malware, and vulnerability research.
13. See GitHub Terms of Service.
14. See GitHub Blog GitHub's developer-first approach to content moderation.
15. See About GitHub Pages.

2.4.1 Case Study 4: youtube-dl

youtube-dl[16] is a popular free software tool for downloading videos on YouTube and other video hosting services. In 2020, GitHub initially took down the youtube-dl repository following a DMCA notice[17] from the Recording Industry Association of America (RIAA), which alleged that the tool violated anti-circumvention provisions under Section 1201 of the DMCA. This led to significant backlash from the developer community both because the software did not circumvent any technological protection measures, and because it had a wide range of legitimate uses, such as for archival, accessibility, educational, and journalistic purposes. The Electronic Frontier Foundation (EFF) intervened[18] on behalf of youtube-dl's maintainers, providing more information about the project to address the claims made in the RIAA's letter, explaining that the way youtube-dl works does not bypass technical protection measures and highlighting its legitimate uses. Following the additional technical information EFF provided and after working with the maintainer to address the allegations of infringement from test examples that referenced copyrighted works, GitHub reinstated youtube-dl.[19] Following this incident, GitHub overhauled its DMCA Section 1201 process so that all circumvention claims will have legal and technical review.

3 Moderating a developer community

Moderating a developer community requires a multifaceted approach that upholds the unique norms and values central to the developer community, particularly within the source community where collaboration and transparency are key. Empowering users to establish clear community norms and integrating feedback from the community ensures that policies and practices remain relevant and responsive to the evolving needs of developers. Effective moderation goes beyond simple content takedowns; it involves nuanced strategies that address violations while preserving critical content. Encouraging community-led moderation empowers developers to take an active role in supporting the health of their own projects and communities, while global moderation strategies must account for the diverse and international nature of the developer community, balancing local legal requirements with the need for global collaboration.

3.1 Developer community norms

The software developer community, particularly among open source developers, is characterized by a unique set of norms and values centered around collaboration, transparency, and shared responsibility. Volunteer maintainers and open source stewards play critical roles in this ecosystem, often dedicating their time and expertise to manage

16. See youtube-dl repository.
17. See RIAA takedown notice in GitHub DMCA repository.
18. See EFF letter to GitHub on youtube-dl.
19. See GitHub Blog Standing up for developers: youtube-dl is back.

projects, review contributions, and ensure the integrity and progress of the software (Geiger, Howard, and Irani 2021). These individuals operate under principles of meritocracy and peer review, where contributions are evaluated based on quality and impact rather than hierarchy. This communal approach fosters innovation and rapid development but also requires a delicate balance of inclusivity, trust, and respect for intellectual property. Open source stewards, in particular, must navigate the complexities of guiding project direction, mediating conflicts, and sustaining community engagement, all while maintaining the open, collaborative spirit that defines the open source movement.

3.1.1 Case Study 5: xz backdoor incident

On March 29, 2024, engineer Andres Freund[20] discovered a backdoor in the widely used xz[21] compression library (Roose 2024). Starting in 2021, a developer using the likely fictitious name Jia Tan (JiaT75), along with other aliases, used social engineering to gain the trust of the maintainer, Lasse Collin, and to be added as a maintainer to the project, eventually inserting a backdoor into versions 5.6.0 and 5.6.1 of xz Utils (Kaspersky Global Research and Analysis Team 2024). This backdoor allowed unauthorized access to systems using the compromised versions, posing a severe threat because of the extensive use of xz in popular Linux distributions. If this backdoor had not been discovered, it likely would have had broader reach than the SolarWinds event of 2020, which affected more than 18,000 customers, including several U.S. government agencies and major technology companies (Goodin 2024, Oladimeji and Kerner 2023).

GitHub's Trust and Safety team was faced with the challenge of needing to act quickly to prevent potentially unknowing users from downloading code with backdoors while recognizing that disabling a widely used project would create additional friction for developers and security researchers performing forensic analysis. GitHub initially blocked both of the maintainers' accounts and disabled the content, and then quickly unblocked Lasse Collin when it became clear that he was not involved in the attack. Once communication with Collin was established, the decision to reinstate the original code was left to him in order to respect the volunteer maintainer's time and autonomy. Collin reinstated the repository and shared notes[22] and an FAQ authored by another developer, Sam James,[23] on the incident.

The incident underscored the vulnerability of critical open source projects to advanced persistent threats, the need for improved security practices within the open source community, and the challenges of volunteers maintaining widely used open source projects that comprise critical digital infrastructure. It also highlights the important role that technology companies play in securing open source; Freund discovered the vulnerability during routine maintenance work as a Microsoft engineer Roose 2024. Moderation actions like those necessitated by the xz backdoor incident should be seen

20. See initial report posted by Andres Freund on openwall March 29, 2024.
21. See xz utils library.
22. See notes on xz backdoor incident shared on Collin's website.
23. See GitHub Gist of xz backdoor FAQ authored by Sam James.

as a last resort, with the technology industry focusing on preventative solutions and supporting the health of the open source ecosystem. To that end, GitHub has established its own funding instruments to support open source work; GitHub Sponsors[24] established a venue for open source maintainers to get funding for their work from consumers or supporters of their open source projects. GitHub also supported the launch of the Open Technology Fund's Free and Open Source Software Sustainability Fund[25] in 2023. Significant industry and government support is necessary to establish robust, long-term open source sustainability support.

3.1.2 Case Study 6: Integrating developer feedback into site policies

GitHub involves users in collaborative development of its site policies, procedures, and guidelines. Users are able to view all changes to terms on the site policy repository,[26] where users can fork, use, and adapt its open-sourced policies for their own purposes as well as provide feedback or suggestions for site policies by opening an issue or a pull request. GitHub first open-sourced its site policies in 2017,[27] and since then the site policy repository has been a means for users to both provide substantive feedback and flag simple errors and typos. The site policy repository provides its own contribution guidelines[28] and code of conduct[29] adapted from version 1.4 of the Contributor Covenant.[30]

GitHub's Terms of Service[31] states that it will provide users with a 30-day notice of material changes to its terms, a policy that was introduced in 2017. GitHub reviews feedback on site policy changes and will engage with users, provide clarification, and update the proposed policy change in response to user feedback. The 30-day notice-and-comment period was most recently used in April 2024 to add a policy on synthetic and manipulated media tools to GitHub's Acceptable Use Policies.[32] In this instance, GitHub staff provided clarification[33] in response to user's questions on the policy addition but did not make changes to the policy itself. Using GitHub as a platform to get feedback on GitHub site policies is an effort to embody the spirit of the developer community, meet developers where they are, and have developers review and make changes in the format they are expecting.

3.2 Platform moderation beyond takedowns

Internet services have a wide range of options beyond takedowns to address user accounts and content that violate the rules (Goldman 2021). As discussed above, there

24. See GitHub sponsors site.
25. See the Open Technology Fund's Free and Open Source Sustainability Fund.
26. See GitHub site-policy repository.
27. See GitHub Blog Open sourcing our site policies and new changes to our Terms of Service.
28. See GitHub site-policy contributing guidelines.
29. See site-policy code of conduct.
30. See Contributor Covenant, version 1.4.
31. See GitHub Terms of Service Section Q. Changes to These Terms.
32. See GitHub Acceptable Use Policies on Misinformation and Disinformation.
33. See GitHub staff comments on site-policy pull request.

are important reasons for GitHub to aim to keep as much code available as possible that complies with the law and GitHub's own Terms of Service and Acceptable Use policies. Code is speech, the foundation of critical digital infrastructure, and a shared endeavor across the developer community. Code that may be used for malicious purposes or for copyright infringement may have societally beneficial dual uses. The network effects of takedowns and the functional aspect of code are important factors of consideration for content moderation on GitHub.

GitHub has developed a suite of content moderation tools over the years in response to specific needs. Overall, GitHub's primary moderation tools take action at the user level: flagging to hide or suspending accounts. Flagging and suspending have different use cases. Flagging hides all of a user's content but allows them to continue using the platform; this tool is typically used for disruptive users or low value/spammy content. Suspending is employed when a user's behavior has violated GitHub's Terms of Use or Acceptable Use Policies; their account is suspended, but their content remains visible on GitHub if it is of use to others.

Table 1: Moderation tools at the user level

Tool	Impact	Typical purpose
Flag	Hides all of a user's content, but allows them to continue using the platform	Hiding disruptive users and spammy or low-value accounts/content
Suspend	Blocks users from being able to log in to their GitHub account, but keeps content on GitHub visible	User behavior has violated GitHub ToS and AUPs, but has made contributions that are valuable to others

Unlike other content platforms, GitHub's Trust and Safety team cannot edit a file or delete a line of code. There are both ethical and practical reasons for this limitation: editing a user's code assumes a level of ownership GitHub does not have and could break a software project in use by other developers. Likewise, individual files within a repository cannot be hidden or disabled. Moderation thus takes place on the repository level. GitHub considers disabling a repository an all-or-nothing decision that should be avoided unless necessary, so it has developed a breadth of moderation actions that can be applied beyond full takedowns. These moderation actions can generally be grouped into tools that limit the visibility or reach of a project, which allows the content to remain on the site but limits the conditions under which that content circulates, and tools that add friction to make content less readily available and to inform users, through content labeling and interstitials, about the nature of the project's content (Gillespie 2022; Morrow et al. 2022). There are also infrequently used tools to temporarily limit interaction on a given project, which are used at the request of a maintainer during a rapidly developing event such as a harassment campaign.

Table 2: Moderation tools at the repository level

Purpose	Tool
Visibility: used to limit where or how many people may discover a project, such as when content violates laws in certain jurisdictions or may be used maliciously	• Disable • Geoblock • Restrict visibility to collaborators only • Exclude from explore on github.com (e.g., do not promote)
Friction: used to make content less readily available and to inform users about the nature of content, such as projects that may contain explicit content or misinformation	• Require users to log in to access • Banner interstitial (dismissible informative banner across the top) • Blocking interstitial (must be logged in and click through to access)
Interaction: applied at request of user to slow rapidly developing events such as harassment campaigns	• Limit to existing users • Limit to prior contributors • Limit to repository collaborators

3.2.1 Case Study 7: GitHub Pages and educational use

As discussed previously, GitHub Pages creates additional moderation challenges for GitHub Trust and Safety because it is intended solely for the use of developers' projects on GitHub, not general-purpose website hosting. A distinct challenge is the use of Pages for educational exercises, wherein a student may be assigned to make a copy of an existing website as a web design or programming exercise. These Pages sites may look like and can be reported as phishing, so GitHub Trust and Safety developed a policy[34] for the use of Pages for educational exercises in December 2023. This policy clarified that the use of Pages to create a copy of an existing website as a learning exercise is not prohibited, but users must write the code themselves, the site may not collect any user data, and there must be a prominent disclaimer on the site indicating that the project is not associated with the original and was only created for educational purposes. There is currently no way for GitHub Trust and Safety to append interstitials onto Pages sites, so GitHub asks users to add disclaimers to their own sites and has introduced a disclaimer interstitial for repositories that contain code for these educational exercise websites. While this does create more moderation work and deliberation than simply removing all sites that are copies of existing websites, GitHub values providing a platform for software development education.

3.3 Community content moderation

GitHub's approach to content moderation emphasizes empowering developers by encouraging project owners and maintainers to set clear expectations[35] through documen-

34. See GitHub Pages policy on educational exercises.
35. See GitHub Docs About community management and moderation.

tation such as README,[36] CONTRIBUTING,[37] and CODE_OF_CONDUCT[38] files. These user-created documents serve as guidelines for collaboration and provide a basis for moderating interactions and contributions. In a 2021 study of code of conduct conversations on GitHub, researchers found that codes of conduct are used both proactively to encourage participation, and reactively to respond to controversial behavior (Li et al. 2021). An estimated ~56% of public, non-fork repositories have a README file, which demonstrates both growing adoption of project documentation and an opportunity to encourage users to provide more information and guidance on their projects. GitHub also offers repository owners and maintainers moderation tools, including user blocking, conversation locking, and comment moderation, and allows repository owners to delegate moderation responsibilities to trusted collaborators.

Table 3: Project documentation of public, non-fork repositories on GitHub, adoption as of 2024-07-01

File type	Purpose	Adoption	Adoption %
README	Describes project and communicates expectations for collaborating on the project	121,653,718	~ 56%
CONTRIBUTING	Explains how people can engage	1,645,352	~ 0.7%
CODE_OF_CONDUCT	Specifies what kinds of contributions or participation are—and are not—allowed there	635,359	~ 0.3%

3.4 Moderating a global developer community

A key aspect of the global developer community is its truly global nature. GitHub's user base spans every region of the world, and its public website is accessible from every country except North Korea. GitHub has advocated to the US Treasury Department's Office of Foreign Assets Control (OFAC) to offer more of its services in sanctioned countries, and was able to secure a license to offer all of its services to developers in Iran in 2022.[39] In its advocacy efforts, GitHub has emphasized that access to code collaboration fosters human progress, enhances international communication, and promotes free speech and the free flow of information. Ensuring worldwide access allows developers in regions with fewer local resources to participate in and benefit from global projects, leveling the playing field and promoting equitable opportunities.

Global collaboration is necessary to address global challenges, as evidenced by Our World in Data's repository[40] of global COVID-19 data, which supported coordinated

36. See GitHub Docs About READMEs.
37. See GitHub Docs Setting guidelines for repository contributors.
38. See GitHub Docs Adding a code of conduct to your project.
39. See GitHub Blog Advancing developer freedom: GitHub is fully available in Iran.
40. See Our World in Data covid-19-data repository.

worldwide medical response and comparative research of policy responses.[41] Cyber threats do not acknowledge borders, and a 2021 report from the Ransomware Task Force recommended a "whole of world" approach to confront digital threats, noting that siloed tactics are less effective than a globally coordinated response (IST 2021). In 2023, GitHub released the GitHub Innovation Graph website[42] an open data and insights platform on global and local developer activity. The Innovation Graph was developed to provide researchers, policymakers, and developers with valuable data and insights into global developer impact to assess the influence of open source on the global economy and demonstrate the interconnectedness and resilience of the global developer community [43]

GitHub's broad global availability creates specific challenges for content moderation. Many countries with growing software developer communities also have governments that impose restrictions on free speech and information sharing. GitHub sometimes receives requests from governments to remove content that is deemed illegal in their local jurisdiction. Government takedown requests require specific information and must be confirmed to have come from an official government agency; an official must send an actual notice identifying the content and specifying the source of illegality in that country. When GitHub removes content under this policy, removal is limited to the jurisdiction where the content is illegal in the narrowest way possible, such as geoblocking content only in a local jurisdiction. GitHub posts the official requests publicly in its government takedown repository[44] and visualizes the takedown data in its Transparency Center.[45]

4 New frontiers for content moderation

The nuances and challenges of moderating a code collaboration platform have driven GitHub to develop tailored approaches to issues like copyright and security research that consider the network effects of takedowns and potential dual uses of software. Over time, GitHub has developed a wide range of moderation tools beyond takedowns to address specific needs, such as limiting visibility and adding friction to access. GitHub's approach has also evolved as a result of notable events, such as the youtube-dl takedown and reinstatement, which led to adding mandatory legal and technical reviews to its DMCA Section 1201 process. Ultimately, GitHub's approach to content moderation is driven by the evolving needs of the developer community. As we consider new frontiers in content moderation for code collaboration platforms, it's essential to consider how emerging technologies will transform code collaboration and how societal changes will influence

41. See Our World in Data Policy Responses to the Coronavirus Pandemic.
42. See Innovation Graph Website.
43. See GitHub Blog New data and visualizations highlight the resilience of international developer collaboration.
44. See government takedown repository.
45. See GitHub Transparency Center government takedowns.

the landscape in which this collaboration occurs.

4.1 AI and code collaboration platforms

GitHub introduced AI tools for the developer platform with its launch of GitHub Copilot, a pair programmer or auto completion tool that makes suggestions in real time as developers work, and is moving toward integrating AI throughout the software development lifecycle,[46] including vulnerability prevention, pull requests, documentation, and the command-line interface. A notable shift is enabling the use of text and verbal commands for code generation, which will make it easier for more people without specialized knowledge to develop software. But as AI-powered tools lower the barrier to entry to becoming a developer, GitHub will encounter new moderation challenges. GitHub continually works to prevent bad actors from proliferating malicious code on its platform and will face new challenges of preventing these actors from exploiting automated systems. Likewise, addressing the sheer volume of AI-generated content while maintaining GitHub's standards for developer-first content moderation creates challenging questions for scalable moderation mechanisms in the AI era.

4.2 GitHub and AI model hosting

GitHub has a dual role within the AI ecosystem as both an AI tool provider (GitHub Copilot) and a "model hub and hosting service" that makes AI model components available to downstream developers (Srikumar, Chang, and Chmielinski 2024). While GitHub is primarily a generic software development platform that did not until recently have the AI-specific features[47] of model marketplaces, such as enabling inference and deployment, developers can use GitHub to upload and share code for models (Gorwa and Veale 2023). As the AI developer community grows, platforms will have to consider risk mitigation strategies to address the challenges of being an intermediary for open foundation models (Srikumar, Chang, and Chmielinski 2024). These strategies, such as establishing content moderation practices for hosted models, enforcing consequences for policy violations, and adhering to transparency reporting standards, are in line with GitHub's ongoing content moderation approach, but they will require specific considerations of the nuances of moderating open foundation models (Srikumar, Chang, and Chmielinski 2024). Considering that AI models are being released under "increasingly complex and atypical software licenses," GitHub may encounter content moderation challenges from model developers who utilize behavioral use licenses that are difficult to enforce (Gorwa and Veale 2023, Contractor et al. 2022). GitHub's moderation practices reflect the norms of internet platforms and the open source community that have developed over several decades. Because the norms for AI development are not as well established, responding to these new frontiers in content moderation will require dynamic and adaptable platform governance.

46. See GitHub Blog GitHub Copilot X.
47. See GitHub Blog Introducing GitHub models.

4.2.1 Case Study 8: Deepfake policy

In April 2024, GitHub requested feedback[48] on a proposed addition[49] to its Acceptable Use Policies on Misinformation and Disinformation to address the development of synthetic and manipulated media tools for the creation of disinformation and nonconsensual intimate imagery (NCII). This policy update was motivated by rising concern over the use of AI-generated disinformation in a significant global election year and several high-profile cases of deepfake technology used to generate NCII (Jeong 2024). GitHub's proposal was modeled after its approach to malware and dual-use security research (as discussed in Case Study 3), wherein GitHub disallows projects that are fine-tuned for abuse or used in active attack but permits valuable research on synthetic and manipulated media tools. During the 30-day notice-and-comment period, GitHub provided clarification to user's questions about the policy update but did not make changes to the policy itself. After the policy went into effect on May 21, some projects were taken down for being specifically oriented toward NCII and disinformation, while other users were given an opportunity to make changes to their project to keep it in line with Acceptable Use policies. This recent update demonstrates adapting policies to reflect new technologies and their impacts while maintaining a consistent approach to dual-use technologies and valuable research.

4.3 Scaling moderation for a growing developer community

In an April 2024 presentation, GitHub CEO Thomas Dohmke[50] made the projection that because AI developer tools will make software development easier and more accessible, by 2030 there will be one billion developers on GitHub. Growing GitHub to become a platform of one billion developers would create significant challenges for content moderation, requiring robust systems to handle increased volumes of code, discussions, and potential misuse. Considering that GitHub has developed content moderation practices that befit the specific nuances and challenges of a code collaboration platform, including applying legal and technical review where necessary, continued innovation will be required to maintain these developer-first standards at scale—including increasing the capabilities of GitHub's Trust and Safety team with AI tools, just as AI tools have increased the capabilities of software developers.

48. See GitHub Blog A policy proposal on our approach to deepfake tools and responsible AI.
49. See GitHub site-policy pull request.
50. See April 2024 TED Talk With AI, anyone can be a coder now.

References

Bounegru, Liliana. 2023. "The Platformisation of Software Development: Connective Coding and Platform Vernaculars on GitHub." *Convergence,* https://doi.org/https://doi.org/10.1177/13548565231205867.

Contractor, Danish, Daniel McDuff, Julia Katherine Haines, Jenny Lee, Christopher Hines, Brent Hecht, Nicholas Vincent, and Hanlin Li. 2022. "Behavioral Use Licensing for Responsible AI." In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency,* 778–88. https://doi.org/10.1145/3531146.3533143.

Dame-Boyle, Alison. 2015. "EFF at 25: Remembering the Case That Established Code as Speech." *EFF* (April 16, 2015). https://www.eff.org/deeplinks/2015/04/remembering-case-established-code-speech.

Geiger, R. Stuart, Dorothy Howard, and Lilly Irani. 2021. "The Labor of Maintaining and Scaling Free and Open-source Software Projects." *Proceedings of the ACM on Human-Computer Interaction* 5, no. CSCW1 (April 22, 2021): 1–28. https://doi.org/10.1145/3449249.

Gillespie, Tarleton. 2022. "Do Not Recommend? Reduction as a Form of Content Moderation." *Social Media + Society* 8 (3). https://doi.org/10.1177/20563051221117552.

Goldman, Eric. 2021. "Content Moderation Remedies." *Michigan Technology Law Review* 28, no. 1 (March 24, 2021): 1–59. https://doi.org/10.2139/ssrn.3810580.

Goodin, Dan. 2024. "What We Know about the xz Utils Backdoor That Almost Infected the World." *Ars Technica* (blog), April 1, 2024. https://arstechnica.com/security/2024/04/what-we-know-about-the-xz-utils-backdoor-that-almost-infected-the-world/.

Gorwa, Robert, and Michael Veale. 2023. "Moderating Model Marketplaces: Platform Governance Puzzles for AI Intermediaries." *Law Innovation and Technology* 16, no. 2 (November 21, 2023). https://doi.org/10.48550/arXiv.2311.12573.

Institute for Security and Technology. 2021. *Combating Ransomware: A Comprehensive Framework for Action: Key Recommendations from the Ransomware Task Force.* Research report. https://securityandtechnology.org/ransomwaretaskforce/report/.

Jeong, Sarah. 2024. "All the AI Disinformation from the 2024 Elections." *The Verge,* July 30, 2024. https://www.theverge.com/policy/24098798/2024-election-ai-generated-disinformation.

Kaspersky Global Research and Analysis Team. 2024. "Assessing the Y, and How, of the XZ Utils incident." *SecureList* (blog), April 24, 2024. https://securelist.com/xz-backdoor-story-part-2-social-engineering/112476/.

Li, Renee, Pavitthra Pandurangan, Hana Frluckaj, and Lauran Dabbish. 2021. "Code of Conduct Conversations in Open Source Software Projects on GitHub." *Proceedings of the ACM on Human-Computer Interaction* 5 (CSCW1): 1–31. https://doi.org/10.1145/3449093.

Milli, Smitha, Micah Carroll, Yike Wang, Sashrika Pandey, Sebastian Zhao, and Anca Dragan. 2024. "Engagement, User Satisfaction, and the Amplification of Divisive Content on Social Media." *Knight First Amendment Institute at Columbia University,* https://doi.org/10.48550/arXiv.2305.16941.

Morrow, Garrett, Briony Swire-Thompson, Jessica Montgomery Polny, Matthew Kopec, and John P Wihbey. 2022. "The Emerging Science of Content Labeling: Contextualizing Social Media Content Moderation." *Journal of the Association for Information Science and Technology* 73 (10): 1365–86. https://doi.org/https://doi.org/10.1002/asi.24637.

Oladimeji, Saheed, and Sean Michael Kerner. 2023. "SolarWinds Hack Explained: Everything You Need to Know." *TechTarget,* https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know.

Penney, Jonathon. 2019. "Privacy and Legal Automation: The DMCA as a Case Study." *Stanford Technology Law Review* 22 (1): 412–86. https://ssrn.com/abstract=3504247.

Reynolds, Matt. 2019. "What is Article 13? The EU's Divisive New Copyright Plan Explained." *Wired,* https://www.wired.com/story/what-is-article-13-article-11-european-directive-on-copyright-explained-meme-ban/.

Roose, Kevin. 2024. "Did One Guy Just Stop a Huge Cyberattack?" *New York Times* (April 3, 2024). https://www.nytimes.com/2024/04/03/technology/prevent-cyberattack-linux.html.

Schaffner, Brennan, Arjun Nitin Bhagoji, Siyuan Cheng, Jacqueline Mei, Jay L. Shen, Grace Wang, Marshini Chetty, Nick Feamster, Genevieve Lakier, and Chenhao Tan. 2024. ""Community Guidelines Make this the Best Party on the Internet": An In-Depth Study of Online Platforms' Content Moderation Policies." In *Proceedings of the CHI Conference on Human Factors in Computing Systems,* 1–16. May 11, 2024. https://doi.org/10.1145/3613904.3642333.

Scott, Stewart, Sara Ann Brackett, Trey Herr, and Maia Hamin with the Open Source Policy Network. 2017. *Avoiding the Success Trap: Toward Policy for Open-source Software as Infrastructure.* Research report. Atlantic Council. https://www.atlanticcouncil.org/in-depth-research-reports/report/open-source-software-as-infrastructure/.

Srikumar, Madhulika, Jiyoo Chang, and Kasia Chmielinski. 2024. *Risk Mitigation Strategies for the Open Foundation Model Value Chain.* Research report. Partnership on AI. https://partnershiponai.org/resource/risk-mitigation-strategies-for-the-open-foundation-model-value-chain/.

Stoltz, Mitch. 2019. "Copyright's Safe Harbors Preserve What We Love About the Internet." *Electronic Frontier Foundation,* https://www.eff.org/deeplinks/2019/01/copyrights-safe-harbors-preserve-what-we-love-about-internet.

Synopsys. 2024. *Open Source Security Risk Analysis Report.* Technical report. Synopsys. https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html.

West, Jeremy. 2019. "An Introduction to Online Platforms and Their Role in the Digital Transformation." *Organisation for Economic Co-Operation and Development (OECD),* https://doi.org/10.1787/53e5f593-en.

Authors

Margaret Tucker is a Policy Manager at GitHub.

Rose Coogan is Online Safety Counsel at GitHub.

Will Pace is Director of Trust and Safety at GitHub.

For inquiries about this paper, contact policy@github.com.

Acknowledgements

The authors acknowledge that this commentary is informed by their experience as GitHub employees. The authors would like to thank Mike Linksvayer, Felix Reda, Kevin Xu, Jesse Geraci, Abby Rieflin, and the GitHub Trust and Safety team for their contributions to this paper and platform governance at GitHub.

Keywords

Content moderation; code collaboration; GitHub; platform policy; developer policy.