Trust and Safety Tooling as a User Experience Challenge

Authors: Joseph Seering, Braahmi Padmakumar, and Martina Di Paola

In M. L. Daniel, A. Menking, M. T. Savio, & J. Claffey (Eds.) (In Press), Trust, Safety, and the Internet We Share: Multistakeholder Insights. Taylor & Francis

Abstract

The design of user-facing safety tooling is a dynamic, interdisciplinary field. In this chapter, we aim to provide a starting point for more sophisticated discussion of creative directions in user-centered safety tooling design. We first present three diverse examples of user-facing safety tool design in practice, focusing on tools from the livestreaming platform Twitch, discussing how each tool demonstrates a distinct but complementary understanding of the needs of Twitch users. We next provide a series of examples of safety tool designs from academic research, drawing lessons from these about how academic research can push the boundaries of user-facing safety tooling to leverage more socially-interconnected approaches. We conclude by discussing next steps for broadening discussion and building connections between research and practice.

Introduction

The field and practice of trust and safety have been characterized over the years through a number of metaphors, ranging from governance to policing to administration (Gillespie, 2018; Klonick, 2017; Seering et al., 2022). These metaphors highlight the important and complex processes taking place within companies to characterize and respond to problematic content and users. However, these metaphors may unintentionally obscure the safety processes happening at the direction of users themselves. By framing trust and safety as something done *to users* or *on behalf of users*, we miss how much users drive their own online safety, often—but not always—through means granted to them by platforms.

In this chapter, we consider trust and safety tooling from the perspective of user experience, focusing on how various tools and features provided by platforms or constructed by technically-savvy third-party developers allow users to shape their own experiences on platforms. This approach does not seek to minimize the important technical work being done to develop tooling used by trust and safety professionals, but

rather notes that, in many cases, users' safety can be shaped significantly by choices they make themselves, on the front end of the platform experience.

In the following sections, we present a series of examples of user-facing trust and safety tools drawn both from modern social platforms and from research. Our goal with this chapter is to show the diversity of possible tool designs and to inspire both increased creativity in future design of safety-related tools as well as to support collaboration between academic researchers, who may be better-positioned to develop and test less-conventional approaches to safety, and practitioners, who are better-positioned to put the lessons learned into practice at scale.

As we note later in this chapter, safety is a field where there is tremendous room for creativity, but an increasing focus on regulatory compliance has stifled the ability of designers and developers to explore approaches that have clear potential but may not lead to improvements in standard safety metrics in the short term (Keller, 2024). We hope that by showcasing examples of creative, user-centered approaches to tooling, we can inspire broader, more ambitious exploration of new futures for online safety and how they can be made manifest through design.

User-facing Safety Tools: Examples from Practice

The design space for user-facing safety tools is broad, and the boundaries between safety and related domains—e.g., privacy—are not clearly defined. While the most core set of tools, e.g., blocking and reporting functions, is commonly agreed upon as a standard part of safety practice, it is useful to consider a wider range of tools both within spaces that are traditionally deemed "safety" and tools from adjacent spaces such as privacy settings or content curation tools. For this chapter, we draw a very loose definition of user-facing safety tools, including all that give users greater control over their social experience for the purpose of facilitating more positive interactions and reducing inter-user harms.

In this section, we investigate three examples of tools that each provide a lesson about how we can better understand safety as a process that is deeply responsive to the needs, habits, and conditions of users. We seek to illustrate the value of a user-centered understanding of safety practices, which can be built upon to inform both the design of features *for* users and to develop infrastructure that allows users to manage their own safety experiences. We draw these three examples from one

platform: Twitch, a livestreaming platform where users ("streamers") generate live content in front of an audience ("chat").[1]

We choose Twitch for two reasons. First, Twitch's structure allows for sophisticated safety interventions at multiple levels. On Twitch, platform-level oversight is complemented by streamer-level community-based controls as well as user-level privacy and safety settings, following a *multi-level* approach to governance (Jhaver et al., 2023). While Twitch maintains the typical enforcement infrastructure — i.e., automated detection combined with a reporting system leading to various types of review — the platform also relies significantly on user-driven community moderation as a first line of defense in a similar manner to Reddit, Facebook Groups, Discord, and others. Twitch streamers and the users they designate as moderators are often very active in shaping the culture and norms of each community according to the streamer's preferences, both proactively through encouraging desired behaviors and reactively by punishing and potentially banning problematic users from their spaces (Seering & Kairam, 2023).[2]

Through a sophisticated suite of user-facing safety features, Twitch provides opportunities for users to shape their safety experiences, but Twitch also provides extensive support for *user-developed* safety features via APIs and access for moderation chatbots (Seering et al., 2018). This approach, where first-party tools are complemented by support for customizable, user-developed safety infrastructure, showcases the breadth of possible tool designs and their inter-related benefits.

It is also useful to situate a discussion of safety features within a context of user needs, and in this case Twitch also provides a clear case study: in the summer of 2021, streamers on Twitch were targeted by a wave of what were termed "hate raids." Though the exact form of these hate raids varied widely, the most common presentation was waves of hateful spam posted in livestream chats, sometimes accompanied by follow-botting. Research studying these hate raids found a high proportion of violently racist and anti-Semitic messages, and noted that many of the targeted streamers were nonwhite and/or transgender (Han et al., 2023). These hate raids received significant media attention, and many streamers from targeted groups publicly criticized Twitch for perceived inaction in responding to these hate raids, leading to an eventual large-scale boycott of the platform on September 1st, 2021.

[1] Information about Twitch tooling provided in this chapter is drawn from academic research and personal experience of the authors and does not reflect any internal or proprietary knowledge about Twitch.
[2] See also Lambert et al., (2024) for a related discussion of moderators' strategies for proactive interventions on Reddit.

The frequency of hate raids decreased significantly over the following months, but a number of platform-developed and user-developed tools were released throughout mid-late 2021 and 2022 that could better respond to this type of attack. Official Twitch feature launches included phone verification and expanded email verification options for accounts, which were announced in late September, 2021, but also a series of more sophisticated features that gave users more fine-grained control over their safety experiences. In this section, we discuss three safety tools that emerged in this period: First, a user-developed tool named Sery_Bot, which became widely used starting in mid-late 2021; second, the Shared Ban Info tool launched in mid-2022, and third, the Shield Mode feature launched in late-2022. Each of these features illustrates a distinct but complementary approach to user-driven safety.

We note here that the purpose of this section is not to evaluate Twitch's response to hate raids; much has been written elsewhere in that regard (Cai et al., 2023; Han et al., 2023; Limbong, 2021). Instead, we focus on this particular platform and time period because it provides a clear context through which we can understand the value of user-centered approaches to safety tooling.

Sery Bot

The relatively sudden spike of hate raids in summer of 2021 meant that immediate responses to hate raids would need to come primarily from users. Users rapidly formed ad-hoc communities to track hate raids, to share advice about how to manage privacy and personal safety settings, and to discuss tools that could be used to protect against hate raids (Han et al., 2023). One notably popular tool was Sery_Bot, a Twitch moderation chatbot developed by a Twitch user and streamer named Sery that was tuned to detect and respond to hate raids.

Sery_Bot was initially a personal project, built on a less urgent timeline starting in 2018 — well prior to the surge of hate raids — to support a variety of types of useful interactions that were not available on other commonly-used bots. Moderation-related features were added later on; one such early feature enabled easy banning of up to 500 users on a shared banlist. This feature was created prior to the hate raids of 2021, but eventually became a useful entry into hate raid related design.

Hate raids escalated in August of 2021, and Sery saw their impact directly within his network of streamers. Despite not being a target himself at the time, he saw an opportunity to help, specifically building on the mass ban feature. He developed a new approach to mass banning based on a common keyword or phrase, responding to the

tendency of hate raid bots to spam specific phrases. Shortly thereafter, he expanded this feature to do some basic automatic detection of hate raids, allowing for greater usability under stressful circumstances. These basic but timely functionalities led to a surge in popularity for the bot.

The rapid expansion of Sery_Bot's user base started to create more challenges as the bot began to hit Twitch's bot rate limits. Sery quickly revised the bot, removing non-moderation related features to focus on the core functionalities necessary to combat hate raids. After a stressful period trying to navigate the rate limits, Sery made direct contact with Twitch staff who helped him get expedited verification for the bot to lift the rate limits. Over the following months, Twitch released various public-facing and behind-the-scenes features for curbing the impact of hate raids, but Sery_Bot's popularity remained high, and Sery has continued making updates and advancements to the bot in the years since. As of the time of writing, Sery_Bot is used in 184,000 streamers' channels.

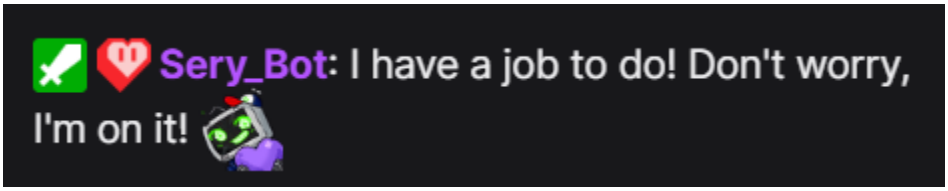

Fig 1. Sery_Bot notification message when the bot is handling a follow-botting incident

Sery_Bot was not intended to replace or compete with any part of Twitch's official safety infrastructure. Rather than attempting to supplant the work being done by Twitch, Sery instead saw an opportunity to leverage the existing infrastructure[3] provided by Twitch to rapidly develop a tool that could protect users while official, first-party solutions were being developed. This user-driven development process highlights several important points: first, it shows the value of direct feedback from users with the greatest safety concerns. Throughout the process of developing Sery_Bot, Sery was able to get direct, rapid feedback from the users who were likely to be the most active users of the tool. Second, the development cycle for Sery_Bot was much more rapid and responsive than for most first-party safety features; if attackers tried a new tactic, Sery could see it in the data and develop and launch a countermeasure within a few days.

Despite the details above, the fact that Sery_Bot has been so widely adopted should not be used as an example of a failure by Twitch. The rapid development time of Sery_Bot was a consequence of its less-formal development process, something that most companies cannot (and should not necessarily aim to) emulate. For example, though

[3] I.e., the support for chatbots via IRC and the API.

the major safety-related features of Sery_Bot were able to be launched quickly, documentation and support for these features took a much less formal approach, often through announcements on Twitter. Similarly, localization remains an ongoing project, and the bot is currently most available in English. Over the years, there have been cases where changes led to unexpected outcomes for users or downtime for the bot; maintenance is always a consideration. For now, Sery_Bot remains fully free to use, though Sery accepts voluntary donations to support its upkeep.

Users were initially drawn to Sery_Bot as a just-in-time, good-enough solution to their immediate urgent problems. They were willing to accept a less-polished experience and higher rates of unintended outcomes due to the seriousness of the immediate situation, and over time they grew to appreciate Sery_Bot as a convenient way to customize their safety experience, but ultimately Sery_Bot was a valuable complement to — not a replacement for — the safety infrastructure on the platform.

Shared Ban Info

Users' responses to hate raids showed how safety in the Twitch ecosystem is deeply community-based and socially-driven. First-line responses to safety crises typically happen within individual communities (i.e., streams), but coordination and safety-related education happen among networks of streamers. Many streamers learn about new safety-related tools and features from friends and connections (Seering & Kairam, 2023), and they turn to their networks for support when circumstances take a turn for the worse. Twitch's Shared Ban Info feature leverages this social structure by giving streamers a way to share information with other streamers about users banned in their communities.[4] When a user who is banned in one channel tries to send a chat in a channel that is connected via the Shared Ban Info feature, their chat message can either be automatically flagged as suspicious or withheld for review before it is seen by other viewers.

Sharing information about banned or blocked users was not a concept invented by Twitch with this tool — it has been implemented across various platforms for many years, albeit usually through user-developed tools (Jhaver et al. 2018). A form of shared blocklists was even present in some of the third-party tools created for Twitch, including Sery_Bot. Twitch's formalization of this feature contributes a few notable additions and some interesting design decisions to the previous iterations of this feature, adding some

[4] https://safety.twitch.tv/s/article/Safer-Together-Making-Twitch-Safer-with-Shared-Ban-Info?language=en_US

social nuance and integrating this feature into a broader multi-level architecture of governance.

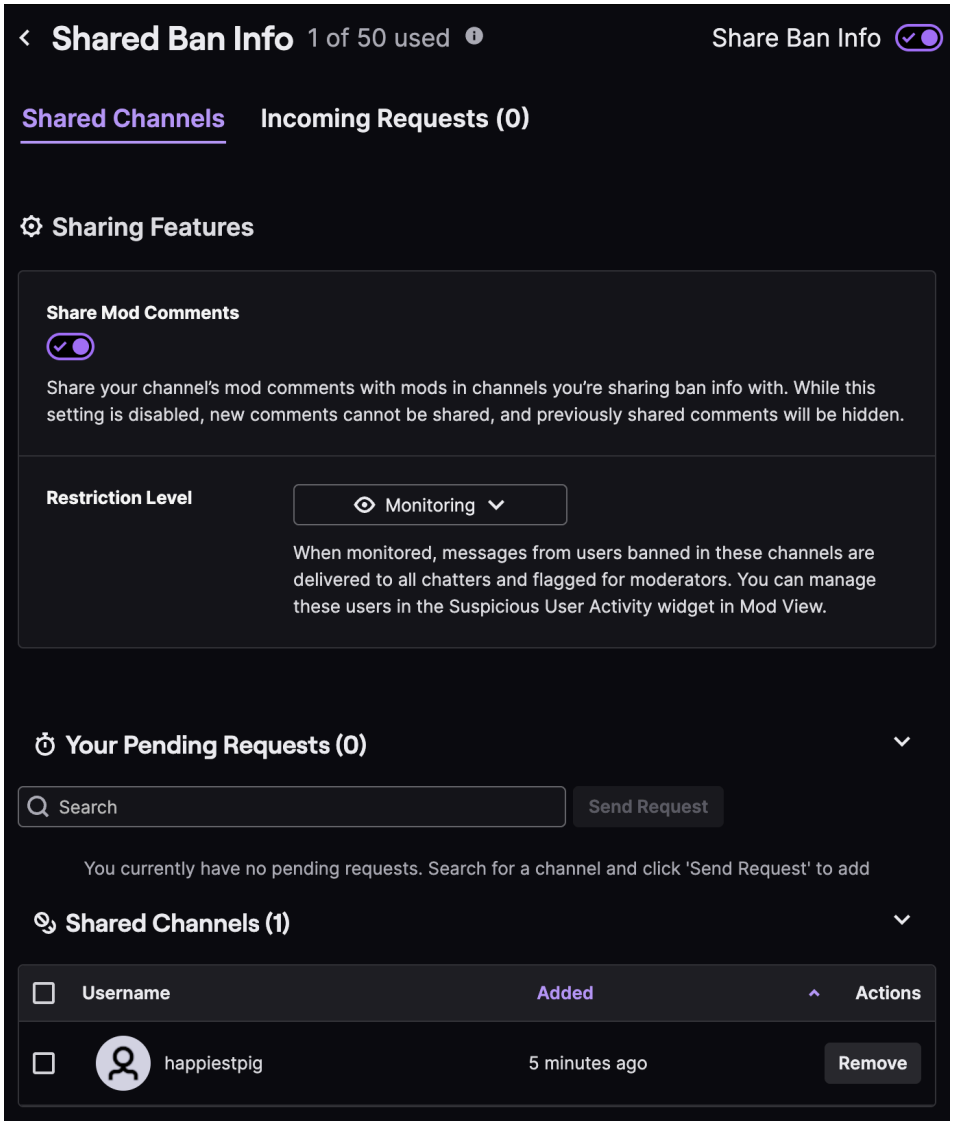

Fig 2. Twitch's Shared Ban Info settings panel

In previous designs, subscribers to a shared blocklist would typically have all users on the list automatically blocked. Twitch's design took a different approach: per the initial launch announcement, there is no option for directly blocking all users blocked in other spaces; they can either be flagged — a minimally invasive intervention called "Monitoring" that only the streamer and their moderators can see — or have their message temporarily withheld pending approval by the streamer or moderators. The streamer and moderators still have the option to ban a flagged user, but this approach takes a lighter touch and acknowledges the problems with previous designs where users who were inaccurately added to a shared blocklist could suddenly find themselves

unable to participate across a breadth of social spaces without any option to appeal (Jhaver et al., 2018).

This design also integrates with other features in the Twitch safety ecosystem, aligning closely with the previously-launched, machine learning driven "Suspicious User Detection" system that was designed primarily to combat ban evasion. As with the Shared Ban Info system, the Suspicious User Detection took a flagging-first approach, withholding or flagging messages from potentially suspicious users but leaving it to the streamer to decide whether or not to ban them. This lighter touch contrasts with the large-scale banning and blocking implemented in Sery_Bot and related user-developed tools, but this difference should not be seen as a misalignment in their understandings of user needs; user-developed tools launched during Hate Raids were responding to urgent, emergency situations, and users were much more willing to accept false positives for user removals under these conditions than they would normally be.

This dichotomy further highlights an important aspect of the Twitch streamer user experience that is related to but somewhat distinct from the experience of other user groups on Twitch and on other platforms: as Twitch is a source of income for many streamers, they can be more hesitant to take aggressive actions in the name of safety that could curtail their channel's growth. Each banned user is a user that will not subscribe or watch ads on the channel, and while the impact of a single ban may not be significant, streamers may have concerns that an overly-aggressive approach to safety may be detrimental to their future revenue and growth. In extreme circumstances, as noted above, this calculus may change, but in the steady state this is a consideration that designers must account for. Twitch's philosophy in this regard is highlighted in the Q&A section on the Shared Ban Info announcement: "You're the expert when it comes to your community, and you should make the final call on who can participate."

Shield Mode

In late 2022, Twitch announced Shield Mode — a one-click toggle that streamers can use to enable a variety of safety settings in their channels quickly in an urgent situation. The metaphor of a shield is clear; the tool provides protection against attack, and the announcement of the feature even specifically references hate raids: "Harassment and hateful behavior can come in waves, such as through a targeted attack, and we hope this tool will make it easier to instantly shut down a hate raid if that ever happens to you."[5]

[5] https://safety.twitch.tv/s/article/Protect-your-channel-with-Shield-Mode?language=en_US

The primary power of Shield Mode lies in its ability to rapidly toggle a channel's settings to a preset emergency combination, tightening moderation settings and restricting who can chat and what can be posted. The tool allows for customization of which settings will be enabled when Shield Mode is activated and at what levels of strictness they will be set. The tool also adds new features, streamlining the mass-banning process and allowing streamers to fully prohibit first-time chatters from participating in their community while Shield Mode is active. As with Shared Ban Info, some of this functionality previously existed in user-developed tools launched during and prior to the wave of Hate Raids in 2021, but Shield Mode formalized and streamlined this process and added new, helpful functionality.

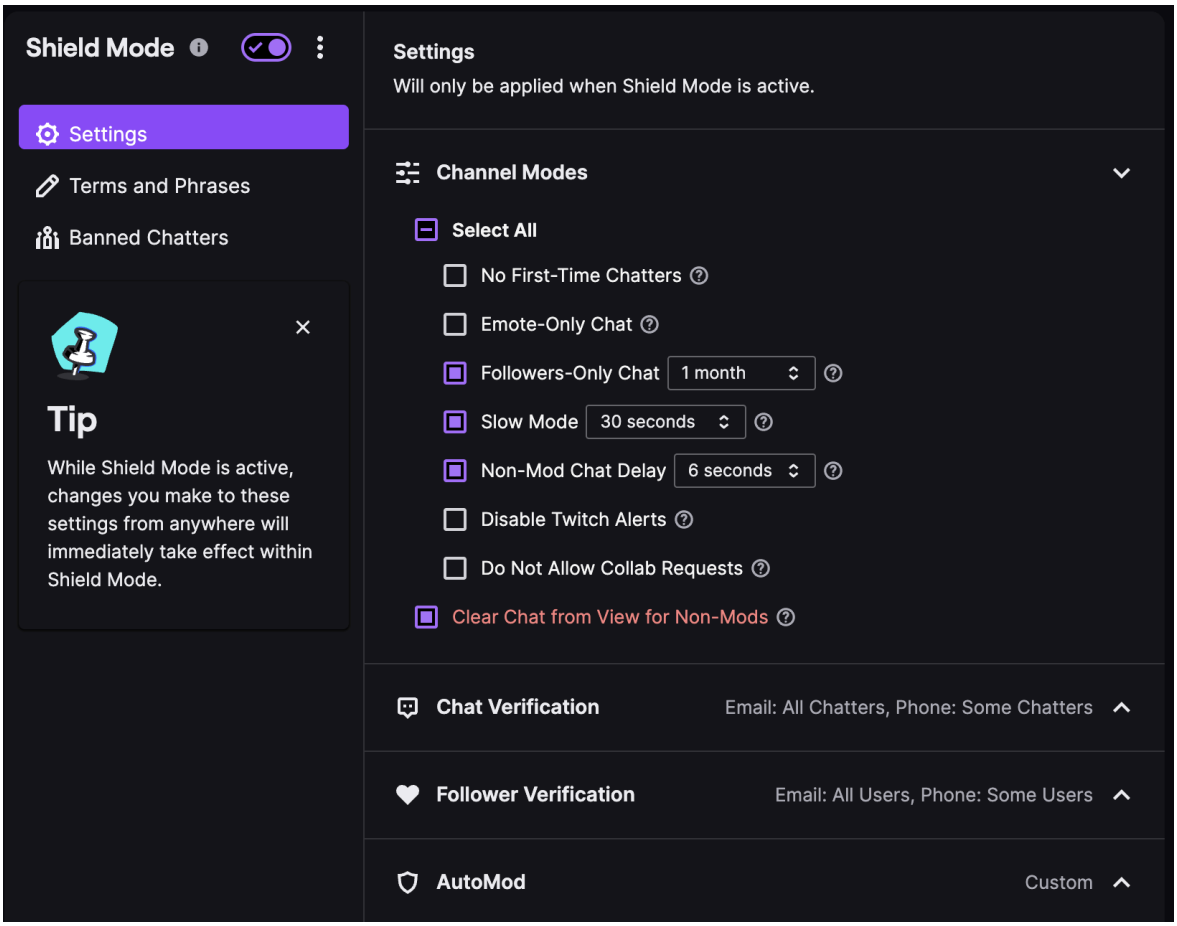

Fig 3. Twitch Shield Mode configuration page

This combination of features is clearly intended for use only in emergencies. Fully prohibiting new chatters from participating in the long term would completely halt the growth of the community, so it would be unusual for a streamer to casually engage this particular feature. With that said, a tool designed for rare use can still be important. In this case, having the ability to quickly lock down a channel provides significant peace of

mind to streamers who are nervous about being the target of a hate raid at some point in the future.

Though Twitch users' reactions to this tool's announcement were significantly positive, designers may face challenges in justifying tool types like this one, as they are unlikely to score well on traditional quantitative usage metrics. If metrics for success rely on frequency of use, the designs of safety-related tools will be significantly constrained to focus on tools that support frequent, mechanical actions. Prior research has critiqued this trend within the design of safety-related tools (Seering et al., 2024), noting that tools that focus on automating common processes like removal and banning have been overemphasized at the expense of tools that support more socially nuanced or rare processes. Shield Mode is an important example of this latter category, presenting a valuable functionality that most streamers will rarely (if ever) use. In justifying this, a useful comparison case is that of the fire extinguisher: in simple terms, most people will never use a fire extinguisher, yet we are glad to have the option to use them should the need arise.

The development of Shield Mode also showcases the importance of learning from users with more extreme needs. Shield Mode was developed with attention to the group of users on Twitch who had some of the most extreme safety needs — namely, streamers from traditionally marginalized groups who are most likely to be targets of harassment in the form of hate raids. While some might note that the needs of populations like these are unrepresentative of the population of users as a whole, this is precisely the value of user experience research in the domain of safety. The majority of users may have relatively few or fairly rare safety needs, so conducting safety-focused UX research on "average" users is likely to be resource-inefficient. Instead, learning from populations with more extreme safety needs can be a much more efficient use of resources because the needs of these users represent a *superset* of the needs of other, more "average" users. Shield Mode presents a clear example of the value of this approach: by learning from the needs of this more extreme population, Twitch created a tool that covers a wide array of safety needs for many different types of users.

Tools in Research

A dynamic body of academic literature, mostly published within the last decade in the field of human-computer interaction (HCI), has investigated the potential for user-facing tool development to support user safety. Explorations of safety tooling can flourish in academic contexts in ways that could be challenging in industry settings, facing less immediate pressure to produce outcomes that conform to existing metrics for success.

This is both a strength and a relative weakness: this type of academic research is typically not conducted with the intent to produce an immediately-usable tool that can be deployed in commercial contexts. Instead it typically aims to make a point about what is possible or how we might reconsider our approaches to difficult problems. Thus, this type of work is best read with a simple question in mind: What is the core concept being advocated for that we hadn't previously considered? In the following sections, we highlight five examples of tools from research that each provide a concept worth considering.

Squadbox: "Friendsourced" moderation

A relatively early example of modern user-facing safety tooling research was *Squadbox*, a tool presented in a paper published in 2018 that allows users to delegate moderation of content to a trusted friend or supporter (Mahar et al., 2018). The core idea of Squadbox is straightforward: "friendsourced" moderation. Drawing from user research with people who faced targeted harassment, they find that such users often rely on friends to help them manage this harassment — distributing the emotional burden and creating distance between hateful content and its intended target — and the paper seeks to outline how such a practice might be supported through design.

The Squadbox tool allows for email moderation, where the trusted friend works to determine which emails the user facing harassment should receive. The system implements various features to support this flow, including sender allow lists and deny lists, controls for evenly distributing work among (potentially) multiple moderators, the ability to customize instructions to moderators, and various automated flags to identify potentially-problematic content. This allows for significant flexibility based on the targeted user's specific circumstances and needs.

However, while the various features and design choices are useful to consider, the core concept of Squadbox is the insight that moderation can be a socially-interconnected practice and that burdens can be reduced if shared. The paper directly discusses whether the "friendsourced" approach simply spreads trauma rather than reducing it, but notes that participants strongly felt that the act of creating distance from harassment significantly reduces its impact.

Today, a number of years after the publication of this paper, systems have been developed that share fundamental elements with Squadbox. For influencers, who are often the targets of much unwanted communication, management companies may act to screen incoming content to prevent direct exposure to harassment. On livestreaming

platforms such as Twitch, certain moderation settings and setups can delegate moderation duties to volunteer moderators, who can act to protect the streamer. With that said, “friendsourced” moderation remains out of reach for most users with more common use cases, despite its clear potential.

Crossmod: Community-driven bootstrapping

Most automated tools in community moderation have relied on fairly simple, rules-based approaches (Jhaver et al., 2019). User-developed moderation bots have traditionally relied on lists of blocked terms or, in more advanced cases, lists of regular expressions, which allow for somewhat more flexibility at the cost of a steeper learning curve (Song et al., 2023). More flexible, machine learning based approaches have typically been out of reach for community moderation for two reasons: first, the technical complexity of developing such a solution in a third-party tool, and second, the difficulty in customizing machine learning based approaches to each community’s individual needs. As shown in a wide variety of academic research and practical deployments, machine learning based approaches are, by definition, least accurate in niche contexts where very limited prior data exists.

Crossmod, a tool presented in a 2019 paper (Chandrasekharan et al., 2019), creatively addresses this latter problem. It begins with the observation that smaller and newer communities may have distinct needs that are not well served by generic machine learning models, but they also lack sufficient prior data to develop custom models from their own past moderation decisions. Crossmod addresses this challenge with the core insight that, while norms vary across communities, each new community will have what could be roughly termed “nearest neighbors” that can provide additional data to draw from.

The core functionality of Crossmod is a “smarter Automod.” It combines decisions from machine learning classifiers trained on past decisions from 100 other communities, as well as from 8 additional classifiers trained for norm-specific evaluation (e.g., identifying homophobic and racist slurs). Moderators from the community deploying Crossmod can set various conditions just as they would with the standard Automoderator tool, e.g., telling the system to flag comments for further review if the system predicts that 80 or more of the 100 communities would have removed that comment, or if another specific community would likely have removed it. Crossmod was an early example of a tool that leveraged the social nature of safety on a community level: communities could benefit from past moderation decisions made by other similar communities, bootstrapping their entrance into ML-driven moderation.

Chillbot and Apolobot: Moving beyond punishment

A third example of research into new approaches in safety tool design comes from a pair of connected papers: *Chillbot* (Seering et al., 2024) and *Apolobot* (Doan & Seering, 2025). These papers each present a bot designed for Discord community moderation with a core, specific purpose: Chillbot provides a quick and easy way for community moderators to privately let users know that they need to "chill," intervening before potentially-problematic situations escalate. Apolobot provides a way to scaffold the process of apology-giving into the community moderation process, where an offender may have their punishment reduced if they sincerely apologize to the person they have hurt. Each of these two tools focuses on one specific kind of interaction in order to test its viability and understand how users might engage with it.

The primary critique that these systems make is twofold: first, that modern approaches to trust and safety have been too quick to jump directly to punishment or removal as a solution to every problem, and second, that these approaches have paid little attention to what happens *after* punishment, ignoring the types of restorative work that are a core part of human social behavior. The papers each tackle one of these two points. Chillbot highlights how *intent matters*, showing that many harms can be prevented without the need for punishment by a just-in-time tailored intervention. For example, many cases of rule-breaking happen simply because the user was not fully aware of the rule (Matias 2019) or because they got into a heated situation and made a mistake. In both cases, the harm could have been prevented if the user had been notified at the right time to pause and reflect on what they were about to do.

Apolobot focuses on the second part of the critique, acknowledging prior literature on restorative justice in online communities (e.g., Xiao, et al., 2022), but aiming for a simpler argument: many of the social spaces users now spend their lives in — the spaces where younger generations will grow up — have been designed with little room for apologies. A heavy focus on developing features for detection and removal or blocking of unwanted content, combined with ease of movement within communities, has removed the necessary social structure. In most cases, it is simply easier for users to choose not to see unwanted content rather than for them to work to foster more positive spaces. In cases where a user has (intentionally or not) caused harm to another user, it is often easier to leave a community and find a new one rather than to work to repair the damage that has been done. (Re)-integrating apologies into online communities may be challenging, but it is worth seriously considering the consequences of a social internet without them.

Post Guidance: Proactive, user-centric, and community-specific feedback

A final example of user-centered safety tooling comes from a collaboration between academic researchers and professionals at Reddit. The Post Guidance feature (Horta Ribeiro et al., 2025) allows subreddit moderators to set rules for posts made in their subreddits, where users may receive a warning and/or be automatically prevented from posting if their post matches a set of predefined conditions. For example, the /r/AskReddit subreddit requires posts to be questions, so the tool could be used to prevent users from posting in that subreddit if their post title doesn't end in a question mark. The paper quantitatively demonstrates that deployment of the tool reduced the workload of subreddit moderators and increased the quality of submitted posts.

This tool is user- and community-centered in that it recognizes the heavy workload that subreddit moderators face and aims to reduce unnecessary burdens to allow moderators to focus on more meaningful work (Schöpke-Gonzalez et al., 2024; Seering et al., 2019). Where moderators of the /r/AskReddit subreddit might otherwise spend a lot of time reviewing flagged posts to manually determine whether or not they contain a question, the Post Guidance tool allows them to shift their focus toward more interesting and important moderation decisions and toward the broader tasks of community development.

This tool is also an important example of a successful collaboration between academics and professionals. The type of large-scale evaluation present in the paper was only possible due to direct involvement of Reddit employees who could develop the tool, recruit users for the study, and analyze resulting data, but the collaboration provides strong evidence supporting the development of similar tools in the future. Direct collaborations between academics and professionals may be less likely to yield provocative, boundary-pushing designs due to the challenges of justifying the work to the host platform, but they are correspondingly more likely to achieve more immediate measurable impact at scale.

Driving a user-centered vision for safety tooling design

With this chapter, we aimed to foster enthusiasm for creative, user-centered design of safety-focused tools. The above examples from both research and practice show the value of thoughtfully designed tools in better serving users' needs and in pushing the

boundaries of how we conceive of what *should* be designed.[6] Ultimately, users' experiences with safety are deeply dependent on their individual and social circumstances. Giving users the opportunity to build a social experience that best matches their situation will lead to a more positive user experience than platform-driven detection and removal in isolation ever can.

Two primary steps will be important for building communities of practice in this domain: first, we call for active exchanges and dialogues between professionals working in this space and their counterparts in academia – both faculty and students who aspire to join professional practice in this field. A more active exchange of ideas can help better disseminate ideas and findings and also guide both research and practice in more productive and creative directions. Second, we call for more direct collaborations between academics and platforms in designing and testing safety tooling. Examples of successful collaborations in this area exist (e.g., Horta Ribeiro et al., 2025; Kim et al., 2022), but are rare.

Safety tooling design is a dynamic, interdisciplinary space, where design will benefit from insights from many fields of study as well as a breadth of technical competencies. It is for this reason that it is an exceptionally promising domain in which academics and professionals can form collaborations and share ideas.

Author Biographies

Joseph Seering is an assistant professor in the School of Computing at KAIST, where he leads the Collaborative Social Technologies Lab. His work focuses on understanding the social and organizational dynamics of digital trust and safety in order to drive the development of new forms of social tools. He has received awards for his papers at the ACM CHI, CSCW, and CHI PLAY conferences, and was named one of the CCC/CRA Computing Innovation Fellows of 2021. He is also an affiliated fellow at the Yale University Social Media Governance Initiative and a member of the Trust and Safety Professionals Association. Prior to joining KAIST, Joseph received his Ph.D. from the Human-Computer Interaction Institute at Carnegie Mellon University, and was a postdoctoral researcher in the Computer Science Department at Stanford University and a fellow at the Stanford Human-Centered AI Institute.

[6] Note that this approach should not be characterized as a form of techno-solutionism, where some set of perfectly designed features will somehow "solve" trust and safety, but rather sees user-centered design as one part of the patchwork of necessary approaches in managing platform safety.

Braahmi Padmakumar is an incoming Master's student in the Collaborative Social Technologies Lab (CSTL) at KAIST. Her interests lie at the intersection of technology and lived experience, with a focus on how online platforms can better support safety, inclusivity, and meaningful social interaction. She is also drawn to the complex interpersonal dynamics shaped by artificial intelligence tools—and to the challenge of designing systems that respond to these nuances with care. She aims to help create social technologies that not only work well, but work well for people.

Martina Di Paola is an undergraduate student in the School of Computing at KAIST. She works as an Undergraduate Researcher at the Collaborative Social Technologies Lab (CSTL) under the mentorship of Joseph Seering. Her research interest focuses on the intersection of social media, content moderation, and feed algorithms, particularly within entertainment platforms. She is committed to exploring how algorithmic curation shapes user experiences and to designing tools that promote positive and engaging online communities.

References:

Chandrasekharan, E., Gandhi, C., Mustelier, M. W., & Gilbert, E. (2019). Crossmod: A cross-community learning-based system to assist Reddit moderators. *Proceedings of the ACM on Human-Computer Interaction*, *3*(CSCW), 1–30. https://doi.org/10.1145/3359204

Doan, B. N., & Seering, J. (2025). The design space for online restorative justice tools: A case study with ApoloBot. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (pp. 1–19). ACM. https://doi.org/10.1145/3706598.3713598

Gillespie, T. (2018). *Custodians of the internet: Platforms, content moderation, and the hidden decisions that shape social media*. Yale University Press.

Han, C., Seering, J., Kumar, D., Hancock, J. T., & Durumeric, Z. (2023). Hate raids on Twitch: Echoes of the past, new modalities, and implications for platform governance. *Proceedings of the ACM on Human-Computer Interaction*, *7*(CSCW1), 1–28. https://doi.org/10.1145/3579609

Horta Ribeiro, M., West, R., Lewis, R., & Kairam, S. (2025). Post Guidance for Online Communities. *Proceedings of the ACM on Human-Computer Interaction*, 9(2), 1-26. https://doi.org/10.1145/3711046

Kim, J., McDonald, C., Meosky, P., Katsaros, M., & Tyler, T. (2022). Promoting Online Civility Through Platform Architecture. *Journal of Online Trust and Safety,* 1(4), 1-23. https://doi.org/10.54501/jots.v1i4.54

Jhaver, S., Birman, I., Gilbert, E., & Bruckman, A. (2019). Human-machine collaboration for content regulation: The case of Reddit Automoderator. *ACM Transactions on Computer-Human Interaction*, *26*(5), 1–35. https://doi.org/10.1145/3330195

Jhaver, S., Frey, S., & Zhang, A. X. (2023). Decentralizing platform power: A design space of multi-level governance in online social platforms. *Social Media + Society*, *9*(4), 20563051231207857. https://doi.org/10.1177/20563051231207857

Jhaver, S., Ghoshal, S., Bruckman, A., & Gilbert, E. (2018). Online harassment and content moderation: The case of blocklists. *ACM Transactions on Computer-Human Interaction*, *25*(2), 1–33. https://doi.org/10.1145/3185593

Keller, D. (2024, October 16). *The rise of the compliant speech platform*. Lawfare. https://www.lawfaremedia.org/article/the-rise-of-the-compliant-speech-platform

Klonick, K. (2017). The new governors: The people, rules, and processes governing online speech. *Harvard Law Review*, *131*, 1598–1670.

Lambert, C., Choi, F., & Chandrasekharan, E. (2024). "Positive reinforcement helps breed positive behavior": Moderator perspectives on encouraging desirable behavior. *Proceedings of the ACM on Human-Computer Interaction*, 8(CSCW2), 1-33. https://doi.org/10.1145/3686929

Limbong, A. (2021, September 1). Twitch users are boycotting over attacks targeting Black, Queer and disabled people. *NPR*. https://www.npr.org/2021/09/01/1032873942/twitch-boycott-hate-raid-attacks

Mahar, K., Zhang, A. X., & Karger, D. (2018). Squadbox: A tool to combat email harassment using friendsourced moderation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (pp. 1–13). ACM. https://doi.org/10.1145/3173574.3174160

Matias, J. N. (2019). Preventing harassment and increasing group participation through social norms in 2,190 online science discussions. *Proceedings of the National Academy of Sciences*, *116*(20), 9785–9789. https://doi.org/10.1073/pnas.1815209116

Mayworm, S., DeVito, M. A., Delmonaco, D., Thach, H., & Haimson, O. L. (2024). Content moderation folk theories and perceptions of platform spirit among

marginalized social media users. *ACM Transactions on Social Computing*, *7*(1–4), 1–27. https://doi.org/10.1145/3632741

Schöpke-Gonzalez, A. M., Atreja, S., Shin, H. N., Ahmed, N., & Hemphill, L. (2024). Why do volunteer content moderators quit? Burnout, conflict, and harmful behaviors. *New Media & Society*, 26(10), 5677-5701. https://doi.org/10.1177/14614448221138529

Seering, J., Flores, J. P., Savage, S., & Hammer, J. (2018). The social roles of bots: Evaluating impact of bots on discussions in online communities. *Proceedings of the ACM on Human-Computer Interaction*, *2*(CSCW), 1–29. https://doi.org/10.1145/3274426

Seering, J., & Kairam, S. R. (2023). Who moderates on Twitch and what do they do? Quantifying practices in community moderation on Twitch. *Proceedings of the ACM on Human-Computer Interaction*, *7*(GROUP), 1–18. https://doi.org/10.1145/3567568

Seering, J., Kaufman, G., & Chancellor, S. (2022). Metaphors in moderation. *New Media & Society*, *24*(3), 621–640. https://doi.org/10.1177/1461444820942097

Seering, J., Khadka, M., Haghighi, N., Yang, T., Xi, Z., & Bernstein, M. (2024). Chillbot: Content moderation in the backchannel. *Proceedings of the ACM on Human-Computer Interaction*, *8*(CSCW2), 1–26. https://doi.org/10.1145/3686941

Seering, J., Wang, T., Yoon, J., & Kaufman, G. (2019). Moderator engagement and community development in the age of algorithms. *New Media & Society*, 21(7), 1417-1443. https://doi.org/10.1177/1461444818821316

Song, J. Y., Lee, S., Lee, J., Kim, M., & Kim, J. (2023). ModSandbox: Facilitating online community moderation through error prediction and improvement of automated rules. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (pp. 1–20). ACM. https://doi.org/10.1145/3544548.3581057

Xiao, S., Cheshire, C., & Salehi, N. (2022). Sensemaking, support, safety, retribution, transformation: A restorative justice approach to understanding adolescents' needs for addressing online harm. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (pp. 1–15). ACM. https://doi.org/10.1145/3491102.3517614