An Exploratory Empirical Study of Trust & Safety Engineering in Open-Source Social Media Platforms

GEOFFREY CRAMER, Purdue University, USA
WILLIAM P. MAXAM III, Purdue University, USA
QIANQIAN LI, University of Rochester, USA
JAMES C. DAVIS, Purdue University, USA

Social Media Platforms (SMPs) are used by almost 60% of the global population. Along with the ubiquity of social media platforms (SMPs), there are increasing Trust & Safety (T&S) risks that expose users to spam, harassment, abuse, and other harmful content online. T&S Engineering is an emerging area of software engineering striving to mitigate these risks. Our study provides the first step in understanding this form of software engineering.

Our exploratory study examines how T&S Engineering is practiced by SMP engineers. We studied two open-source SMPs, Mastodon and Diaspora, which comprise 89% of the 9.6 million open-source SMP users. We focused on the Trust & Safety (T&S) design process, analyzing T&S discussions within 60 GitHub issues. We applied a T&S discussion model to taxonomize the T&S risks, T&S engineering patterns, and resolution rationales. We report that T&S issues persist throughout a platform’s lifetime, they are difficult to resolve, and engineers favor reactive treatments. We integrate our findings by mapping T&S engineering patterns onto a general model of SMPs, to give T&S engineers a systematic understanding of their T&S risk treatment options. We conclude with future directions to study and improve T&S Engineering, spanning software design, decision-making, and validation.

Additional Key Words and Phrases: Empirical software engineering, Social media platforms, Trust & Safety engineering, Engineering decision-making, Risk

1 INTRODUCTION

Social Media Platforms (SMPs) are used by almost 60% of the global population [15]. SMPs enable users to share information, express opinions, and be entertained [80], among other benefits [20, 53]. There are also many documented harms of SMPs, including cyberbullying [8], sexual harassment [76], and online radicalization [45]. Many SMPs rely on manual and automated moderation [61], balancing competing requirements including discourse, preserving the platform’s trustworthy reputation, and keeping users safe.

SMPs are thus at the epicenter of an emerging engineering discipline called Trust & Safety (T&S) Engineering. The Trust & Safety Journal defines T&S as “the study of how people abuse the Internet to cause real human harm” [19]. GitHub defines T&S Engineering as “software designed with user safety in mind” [23]. If we better understand how SMPs can be designed to promote trust and safety, we will help software engineers improve human interactions worldwide. Researchers have previously investigated SMP problems [22, 50, 67] and potential solutions [2, 34, 36, 52]. No literature describes T&S Engineering for SMPs in practice.

In this paper, we describe the first empirical study of T&S Engineering in SMPs. Our goal was to characterize the T&S engineering design process. In particular, we wanted to learn what T&S risks are identified in which SMP features, what solutions are explored, and what properties are prioritized in solutions. We analyzed 60 T&S-related issues from two open-source SMPs, Mastodon (7,833,218 users) and Diaspora (740,409 users) [66]. To do this, we first sampled T&S issues using keywords. Then, we mapped the T&S engineering design process onto a discussion model. Finally, we analyzed elements of this discussion model: risks, treatments, and rationales.

We used a mix of open- and closed- coding to develop taxonomies for the T&S risks, engineering patterns, and pattern selection rationals in SMPs. We used inter-rater agreement to validate our results. We found that T&S issues
remain persistent throughout an SMP’s lifetime. Most T&S issues highlight design shortcomings, not implementation errors. T&S issues are difficult to resolve or remain open, with an average resolution time 147 days longer than other issues. When SMP engineers make a design change to improve T&S, their selected treatments are mostly reactive — their preferred approach is to place the burden on moderators (“Add moderation”) and users (“Require consent”). Although many characteristics of T&S issues are similar between Mastodon and Diaspora, the Mastodon community is more concerned about T&S risks related to toxic content, while Diaspora is focused on privacy issues.

Our contributions are:

- We describe the first study of T&S Engineering.
- We taxonomize T&S risks and threat actors (Table 4), providing researchers and practitioners a useful starting point for future work on T&S risk mitigation.
- We taxonomize T&S decision rationales (Table 6), adapting prior work to the T&S engineering context.
- We taxonomize T&S Engineering patterns (Table 5) and the contexts under which they operate (Figure 4), giving prior grey literature an empirical basis.

Significance: Trust & Safety Engineering is an emerging focus for software engineers whose systems facilitate human interaction. Social Media Platforms are the most prominent such systems. Our work provides the first characterization of the T&S engineering design process for SMPs. Our methodology demonstrates a novel analysis of risk-based decision-making in software engineering. We develop taxonomies for risks, treatment patterns, and decision rationales in T&S engineering discussions. Using these taxonomies, we make empirically-based recommendations for how T&S engineers can implement more trust and safety into SMPs.

2 BACKGROUND

Fig. 1. SMP context diagram showing a one-way interaction between Alice & Bob. Alice interacts with features. Her interactions pass through filters and moderation oversight before reaching Bob. Our study’s focus is highlighted in pink.
2.1 Social Media Platforms: Definition & Types

Hopkins defines the many forms of SMPs [12] comprehensively: “Internet-based... and persistent channel[s] of mass personal communication facilitating perceptions of interactions among users, deriving value primarily from user-generated content” [25]. Smith divides SMPs into 7 building blocks: identity, presence, relationships, reputation, groups, conversations, and sharing [62], each requiring design [35]. These concepts take many forms in SMPs, e.g., user-generated content spans hypertext (e.g., Facebook), video (e.g., YouTube), and photographs (e.g., Instagram). SMPs are among the most popular services on the Internet: over half of the 20 top most visited websites being an SMP as of May 2022 [1].

Many SMPs are operated by companies, introducing potential conflicts between profit and safety, exemplified by the change of ownership of Twitter in 2022 [47]. Open-source (OSS) SMPs try to address this concern. OSS SMPs emerged in 2010 in projects such as Diaspora, pump.io, and GNU Social [82]. Most OSS SMPs are decentralized. In a decentralized SMP, an administrator can deploy an SMP instance on a server for public or private use. Content can be shared across SMP instances through activity stream protocols [81], creating a “Fediverse” (federated universe) [43].

Figure 1 provides a simplified model of SMPs, focused on how platform actors (software engineers, moderators) influence user interactions. User-generated interactions are any action that other users can see. This term encompasses user-generated content and includes posted content, replies, likes, reactions, reshares, quotes, user mentions, private messages, etc.

2.2 Trust & Safety and its Engineering

According to Cryst et al., discussions of Trust & Safety originated in the financial sector in the 1990s to address issues such as fraudulent activity [19]. Platform operators want users to trust the platform and feel safe on it, both in terms of their interactions with the platform provider (e.g., not having their data exploited [18]) and in terms of their interactions with other users (e.g., not being spammed or exposed to harmful content) [73]. Over time, it became clear that any digital platform where users interact will experience T&S issues. Efforts to promote T&S were initially distributed across teams, making it difficult to consolidate best practices and apply research findings [11, 44]. These shortcomings prompted centralization: dedicated “Trust & Safety” teams charged with internal platform governance. Professionalization followed: the Trust and Safety Professional Association (TSPA) launched in 2020, with founding organizations including many SMPs (e.g., Facebook, Twitter, Instagram, YouTube, and OKCupid) [11]. Concurrently, academics at Stanford founded the Trust & Safety Journal in 2021 [19].

Trust & Safety Engineering emerged as a discipline of software engineering in recent years. The goal of T&S Engineering is to consider T&S throughout the software development lifecycle, spanning requirements, design, implementation, validation, and operation (e.g., moderation). We are not aware of prior academic literature that describes T&S Engineering. However, many companies employ T&S Engineers. GitHub says their T&S Engineers “design [software] with user safety in mind” [23] and Leong discusses community safety checks in GitHub release pipelines [42]. GitLab, Cloudflare, and Pinterest advertise T&S Engineering teams [17, 58, 70, 83]. The Trust & Safety Professional Association job board lists many T&S opportunities calling for software engineering experience [72].

Our study applies the concepts of T&S and T&S Engineering. We report the first examination of T&S Engineering in practice.
2.3 T&S in SMPs: A Risk Management View

T&S issues on SMPs are a global challenge. For example, a 2021 Pew Research Center survey of Americans found that ~40% of respondents had experienced online harassment [76]. In 2022, as part of a United Nations action, several nations launched an effort to address online abuse such as on SMPs [26].

To scope the broad definition of T&S to our study of SMPs, in this work we define: User T&S in SMPs as the study of how users harm other users on SMPs, and User T&S Engineering in SMPs as software engineering methods that use knowledge of T&S to reduce harmful user-to-user interactions on SMPs. We use “T&S in SMPs” as shorthand for both concepts, and let context distinguish them.

To organize prior research on T&S in SMPs, we apply the risk management framework from ISO 31000:2018 [29]. We focus specifically on the risk assessment and risk treatment stages of the framework. These are the stages that most directly involve engineers, and which are necessary even if other stages are omitted.

Other researchers have also described T&S challenges in SMPs using risk frameworks [3, 40, 68].

2.3.1 Risk Assessment. The risk assessment step spans the identification, analysis, and evaluation of risks, threats, and vulnerabilities.

Many sources have taxonomized T&S risks and threats on social media [22, 24, 40, 41, 79]. Hasib provided a foundational treatment of SMP risks, considering categories such as traditional information security (e.g., spam, XSS), identity (e.g., phishing, fake profiles), privacy (e.g., digital dossiers, facial recognition), and social threats (e.g., stalking) [24]. Laorden et al. used a threat modeling approach to SMPs to identify additional threats such as private information disclosure and corporate secrets theft [41]. Other researchers expanded these taxonomies, adding categories such as child-specific threats [22], privacy threats such as deanonymization and location leakage [40, 79], and political threats such as disinformation [78]. Thomas et al. provided the most recent and exhaustive taxonomy, enumerating myriad forms of online hate and harassment [67].

Beyond taxonomies, researchers have investigated individual threats. For example: Trabelsi & Bouafif described abuses of content reporting systems [69]; Ashktorab and Vitak investigate cyberbullying mitigation and prevention techniques [8]; Usmani et al. analyze social insider attacks [74]; Such et al. investigated privacy conflicts in co-owned photos [65]; and Cheng et al. studied the efforts of “trolls” to disrupt constructive discussion [16].

Due to its recency and sound methods, we view Thomas et al. [67] as the state-of-the-art taxonomy of T&S risks. We build on it, identifying two additional categories and extending a third.

2.3.2 Risk Treatment. In the risk treatment step, T&S engineers identify candidate treatments to mitigate risks. Two kinds of approaches are used to mitigate T&S risks on SMPs: design and moderation. Figure 1 illustrates these protection mechanisms.

Design Treatments are largely proactive, preventing T&S issues before they manifest. Some SMP design approaches to promote T&S have been investigated in the literature. A set of solutions from Fire et al. [22] include authentication mechanisms, security & privacy settings, internal protection mechanisms, and user reporting features. Some work proposes using the social graph of the SMP to provide context-aware access control, limiting exposure to certain content

---

3This definition excludes T&S issues in the user-platform relationship, e.g., issues about GDPR. There were relatively few such issues in the studied open-source software (OSS) SMPs, perhaps because OSS SMPs lack the profit motivation that leads some commercial platforms to violate T&S. We omitted them during our sampling process.

2ISO 31000:2018 is now behind a paywall. We summarize relevant content here.

The other stages are stakeholder communication, scoping, monitoring, and reporting. These stages are oriented toward engineering leadership and management, and could be omitted by some organizations.
or users [32, 51]. A recent study proposes to change SMP architecture to influence end-user behavior, making it possible to remind users of platform guidelines before posting certain content [36]. Finally, grey literature from Koscik [38] identifies a taxonomy to address abuse vectors with a set of solution patterns.

**Moderation Treatments** are reactive, limiting the impact of problematic user behaviors after they have occurred. For example, in Figure 1, moderation can only apply after Alice interacts with a feature, possibly before Bob sees her behavior. SMP moderation is carried out by platform administrators and automated systems. In many SMPs, moderation is manual, by volunteers or T&S teams [61]). Some platforms moderate automatically, including via per-user and community-based approaches [33].

**Policies**, both external and internal, may influence an SMP’s approach to T&S. *External* policies such as the European Union’s General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA) promote T&S by regulating how organizations can access and process their users’ personal information. Since SMPs derive value from user-generated content (§2.1), such policies affect SMP designs [9]. Additionally, many SMPs have *internal* platform policies developed by platform governance teams, including T&S teams [71]. These policies commonly describe acceptable user behavior (e.g., codes of conduct) and may impact both system design and moderation.

Due to its novelty and comprehension, we view the abuse vector solution taxonomy of Koscik [38] as the most relevant work in the T&S Engineering field for addressing T&S risks. We build on it by identifying five additional categories to treat T&S risks.

### 2.3.3 Risk-based decision-making

To select among candidate risk treatments, the ISO 31000:2018 standard outlines steps to perform risk-based decision-making. They are: (1) risk identification, (2) formulation of risk treatment options, and (3) rationalization and selection of risk treatment plan. The standard indicates six general approaches for a risk treatment: eliminating the activity that gives rise to the risk; increasing the risk to pursue an opportunity; removing the source of the risk; changing the likelihood of the risk; changing its consequences; acknowledging but retaining the risk; and most notably for our study, *sharing the risk* among more parties so that each party faces less risk.

This risk-treatment-rationale model for decision-making is consistent with more general theories of argumentation used in the software engineering research literature [46, 77]. It permits us to build on the state of the art taxonomies for risks [67] and treatments [38] for T&S in SMPs. Since prior work has not considered T&S Engineering specifically, there is no specialized taxonomy for rationales. Among general software engineering rationale taxonomies, we found the rationale taxonomy of Al Safwan & Servant too fine-grained for this purpose [55], and instead contextualized the taxonomy developed by Ko et al. [37].

### 2.4 Summary and Unknowns

SMPs have a significant impact on society. Existing work takes a user-centric perspective in taxonomizing T&S threats in SMP threats, and an algorithmic view of treatments. We know little of the practice of T&S Engineering and risk-based T&S decisionmaking.

### 3 RESEARCH QUESTIONS

Establishing effective T&S engineering practices for SMPs is critical to mitigating the widespread risks that have been discussed. Our research provides initial steps toward achieving this goal.

**RQ1** What are the characteristics of T&S issues?

**RQ2** Risk identification: What risks are identified in T&S issues?
Table 1. OSS SMP projects with over 100K users: user count, GitHub issues, GitHub stars as of January 26, 2023. We studied Mastodon and Diaspora, the top two by all counts.

<table>
<thead>
<tr>
<th>Project</th>
<th>Category</th>
<th>Users [66]</th>
<th>Issues</th>
<th>Stars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mastodon</td>
<td>Microblogging</td>
<td>7,833,218</td>
<td>8,892</td>
<td>39.7K</td>
</tr>
<tr>
<td>Diaspora</td>
<td>Social networking</td>
<td>740,409</td>
<td>4,719</td>
<td>13.2K</td>
</tr>
<tr>
<td>PeerTube</td>
<td>Video sharing</td>
<td>288,964</td>
<td>4,386</td>
<td>11.4K</td>
</tr>
<tr>
<td>pixelfed</td>
<td>Photo sharing</td>
<td>150,326</td>
<td>1,702</td>
<td>4.5K</td>
</tr>
<tr>
<td>Pleroma</td>
<td>Microblogging</td>
<td>127,861</td>
<td>2,983</td>
<td>123</td>
</tr>
<tr>
<td>BirdsiteLive</td>
<td>Microblogging</td>
<td>101,188</td>
<td>91</td>
<td>398</td>
</tr>
</tbody>
</table>

RQ3 Risk treatment: What treatment options are proposed in T&S issues? How are they selected?

4 METHODOLOGY

This study employs a repository mining method [60] to extract and analyze T&S discussions related to OSS SMPs. These repositories have thousands of issues (Table 1), many of which involve T&S topics such as privacy and harassment. Since our study is exploratory, mining repository data provides a cost-effective starting point to identify open challenges in T&S Engineering for future study.

Our mining approach has three steps. (1) We selected popular OSS SMPs. (2) We identified their T&S issues via keywords. (3) We analyzed T&S issue dialogues following a discussion model (§4.3.1) and then coded the model elements for T&S themes and practices.

RQ1 is answered by T&S issue metadata. We determine: when they appear over time, which SMP features they occur in, which phase of the software development lifecycle (SDLC) they involve, and how long they take to resolve.

RQ2 is answered with T&S risk and threat actor taxonomies, based on the risk statements in our discussion model.

RQ3 is answered with taxonomies of (1) T&S engineering patterns, and (2) T&S treatment rationales, developed from treatment option and rationale statements in the discussion model. Because rationales were only coded for closed T&S issues, we split the rationales based on the issue result of merged and no action.

4.1 Repository Selection

To select the specific OSS SMP projects for our study, we consulted an aggregated dataset of all such platforms [66] — see Table 1. Our goal is to study T&S at scale and produce generalizable results. By selecting Mastodon and Diaspora, we can study 89% of the OSS SMP user base and achieve our goal. Both projects use GitHub and track issues via “GitHub Issues” [21, 48].

4.2 Issue Selection

We used a keyword approach to find GitHub Issues containing T&S risk statements. Issue selection followed three phases: selecting baseline keywords, tailoring keywords to the studied projects, and sampling issues. A summary is given in Table 2. One author carried out this process with oversight from another author.

4.2.1 Baseline keywords. We got a baseline set of T&S keywords by aggregating all keywords from the 15 articles in the first two issues of the Trust & Safety Journal [30]. We removed 43 entries unrelated to our definition of T&S in
Table 2. SMP filtering results, summarizing resulting keywords, precision and recall in final batch of keyword expansion, number of T&S issues after the selection process, and proportion examined to reach 30 issues per project.

<table>
<thead>
<tr>
<th>Project</th>
<th>Keywords</th>
<th>Prec., Rec.</th>
<th># T&amp;S Issues</th>
<th>Analysis %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mastodon</td>
<td>17</td>
<td>50%, 100%</td>
<td>431</td>
<td>26%</td>
</tr>
<tr>
<td>Diaspora</td>
<td>15</td>
<td>27%, 100%</td>
<td>316</td>
<td>73%</td>
</tr>
</tbody>
</table>

SMPs (e.g., “robust hashing”), leaving 12 keywords. We used stemming and regular expressions to capture keyword variations. This step reduced Mastodon from 6,523 issues to 659 and Diaspora from 4,699 issues to 182. We applied an additional filter that issues should have at least 5 comments to ensure adequate discussion. This filter reduced Mastodon from 659 issues to 317 and Diaspora from 182 issues to 113.

4.2.2 **Keyword Tailoring.** Next, we tailored the keyword list to each selected repository. Our goal was to find as many T&S discussions as possible. We iteratively sampled 100 issues at a time on each of the two platforms. We added additional keywords in each round based on the T&S in SMPs definition (§2.3). We continued until the recall rate reached 90%. This step expanded Mastodon from 317 issues to 431 and Diaspora from 113 issues to 316.

4.2.3 **Issue Sampling.** Finally, the issues that matched our keywords and passed our filters were randomly sorted for processing. We applied additional filters during this step: (1) Relevance based on the T&S in SMPs definition; (§2.3) and (2) discarding issues marked as duplicates. While processing issues, we found that issues with many comments were overwhelming to model (§4.3.1), so we also filtered out issues with ≥20 comments (13 issues across both projects). We processed issues until a sample size of N=30 was reached in each repository (60 total). This stopping point was chosen due to resource constraints, but was sufficient to expand the state-of-the-art taxonomy in each dimension we examined.

4.3 **Issue Analysis**

After collecting issues, we defined the unit of analysis to be every sentence of every comment including the initial proposal. The resulting issues were analyzed as follows.

4.3.1 **Discussion Modeling.** Analyzing issue discussions is challenging due to their unstructured nature [7, 75, 77]. Our study was focused on the T&S design process, so we modeled the discussions using the risk-based decision-making process described in §2.3.3. This model considers that an engineering decision requires treatment options, their associated risks, and rationales for choosing among them. We modeled each issue accordingly, discarding sentences that did not fall into any of these categories.

**Risk** We label risk identification statements if they contain a T&S risk claim, defined as: the potential loss an SMP faces from users harming other users. This category follows the Risk Identification step of the ISO 31000 standard [28].

**Treatment Option** We label risk treatment option statements if they advance the issue towards closure (e.g. suggesting an implementation or proposing to take no action).

**Treatment Selection Rationale** We label treatment options as chosen if they are accepted by developers and label the treatment selection rationale for accepting or rejecting them. We only coded rationales for chosen treatment options because many unchosen ones did not have sufficient rationale claims, and because justifications for chosen treatment options are most important.

---

4 The baseline keywords are: moderation, suicide, self harm, fake news, misinformation, hate speech, harassment, governance, abuse, safety, cyberbullying, deepfakes.
Figure 2. T&S discussion model for Mastodon issue #9791. The issue proposes an appeal process for moderator decisions. Model elements are numbered by order of appearance in discussion. The initial treatment suggestion #1 is rationalized in #2, a use case. Risks are raised (#3-4). Rationale #5 asserts the treatment would discourage problematic behavior. #6 is a treatment refinement with an associated risk (#7). Treatment #8 addresses risk #3. Green boxes show selected treatment.

Figure 2 depicts the model for one issue. Shapes represent discussion elements. Dashed lines indicate theoretical relations between elements [46, 77]. In practice we found relations challenging to code in these informal design discussions — we eventually omitted them due to irreconcilable inter-rater disagreement.

4.3.2 Development of Taxonomies. A subsequent round of thematic coding was performed across issues and discussion model categories. Most taxonomies leveraged existing literature to better contextualize our work. The base taxonomies are:

- SMP feature list was developed from overarching issue topics and was openly coded (Table 3).
• T&S risk, threat actor taxonomies were developed from risk statements. The T&S risk taxonomy leveraged existing work from Thomas et al. [67] and was assigned based on the T&S risk definition (§4). The threat actor taxonomy was developed from the basic user roles of OSS SMPs and extended when common actors were identified during coding. Assignment followed the threat actor definition (§4).
• A T&S Engineering pattern taxonomy was developed from treatment option statements that treat a T&S risk (Table 5). It extends work from Koscik et al. [38]. Annotators started with this taxonomy and iteratively developed new categories for statements that did not fit.
• A rationale taxonomy was developed from rationale statements (Table 6), leveraging existing work [37]. Specifically, we chose the software quality taxonomy from Ko et al. because we wanted to capture the desired system properties that influenced decisions.

4.3.3 Inter-rater Agreement. Agreement was achieved between two independent annotators coding independently. Agreement was measured using Cohen’s Kappa coefficient [49];

(1) The primary annotator coded each issue, developing the codebook using the processes in §4.3.1 and §4.3.2.
(2) For §4.3.1, coded statements from 10% of the T&S issue discussions were provided to a secondary annotator, who independently coded the statement. The Kappa coefficients for each type code are: 0.89 for risk, 1.0 for treatment option, 1.0 for rationale, and 1.0 for chosen.
(3) For §4.3.2: (1) No agreement process was performed for SMP feature list due to the straight forward nature of the list. (2) For the risk statements, a low Kappa score convinced both annotators to independently code all statements and then resolve disagreements. (3) For the pattern taxonomy, a Kappa score of 0.73 was achieved. (4) For the rationale taxonomy, a Kappa score of 0.81 was achieved.
(4) Based on the “substantial” agreement between the independent annotators on most elements of our analysis, we used the single annotator’s results for the remaining 90% of the data for all elements except the risk statements.

5 RESULTS

5.1 RQ1: What are the characteristics of T&S issues?

We performed a metadata analysis of T&S issues to study the context in which these issues arise. Specifically, we determine when and where they appear over time, which SMP features they present in, and what phase of the SDLC they involve.

Figure 3 displays the percent of all issues and T&S issues created relative to their respective populations. Both Diaspora and Mastodon saw T&S concerns rise roughly 1–2 years after their respective creation dates with continued persistence over time. More than 90% of T&S issues were feature requests rather than bugs.

Table 3 shows the involved platform features and their frequency. The moderation and content sharing features appeared most frequently, followed by user registration. Each feature was also categorized into an element from Smith’s honeycomb model (identity, presence, relationships, reputation, groups, conversations, and sharing) [62]. Note that we added the infrastructure element to account for internal features that users do not interact with.

There are some noteworthy differences by platform in each feature’s T&S involvement over time. One year after Diaspora’s creation, there were a significant number of content sharing T&S issues, indicating that this feature posed many T&S risks to the system. In contrast, the early T&S concerns in Mastodon were moderation, content filters, and instance filters issues. The frequency of user registration issues remained consistent over time, indicating recurring T&S issues in this feature in both platforms.
Fig. 3. Proportion of issues created over time, by SMP (orange–Mastodon; blue–Diaspora) and by type (solid–sampled T&S issues; dashed–all issues). The “all issues” (dashed) and T&S lines (solid) have similar reliability growth curves [64], but the T&S trend seems delayed by 1-2 years.

Table 3. SMP feature list. Features were openly coded per issue after all T&S issues were gathered. Determinations were based on the primary functionality involved in the issue discussion. Element taxonomy follows Smith [62], with additions in bold.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Element(s) [62]</th>
<th>Description</th>
<th>Diaspora</th>
<th>Mastodon</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moderation</td>
<td>Infrastructure</td>
<td>Moderators: monitor content and enforce platform guidelines</td>
<td>4</td>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td>Content sharing</td>
<td>Sharing</td>
<td>Users: post content for others to see</td>
<td>9</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>User registration</td>
<td>Identity</td>
<td>Account creation, verification, and on-boarding</td>
<td>6</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>Private messaging</td>
<td>Conversations, Groups</td>
<td>Users: Direct communication between two or more users</td>
<td>3</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>Content tagging</td>
<td>Sharing</td>
<td>Users: apply labels to their content for discoverability</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>User relationships</td>
<td>Relationships</td>
<td>Users: follow/friend other users</td>
<td>4</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Content filters</td>
<td>Sharing</td>
<td>Users: hide unwanted content</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>User filters</td>
<td>Presence, Relationships</td>
<td>Users: prohibit or ignore communication with other users</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Instance filters</td>
<td>Groups</td>
<td>Users: prohibit or ignore communication with other instances</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Content metadata</td>
<td>Sharing</td>
<td>Users: attach metadata to posted content</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>User profile</td>
<td>Identity</td>
<td>Users: create a page about themselves</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Finding 1: Both projects see T&S issue frequency rise 1–2 years after project creation. The moderation, content sharing, and user registration features are most commonly discussed in T&S issues. The content sharing and moderation features saw respective peaks in activity in 2011 and 2017-2019, respectively. 92% of T&S issues were feature requests instead of bugs.

5.2 RQ2: Risk identification: What risks are identified in T&S issues?

We analyzed the threat actor that each risk statement implicated. Among them were user, moderator (which includes content moderators and server administrators), bot, and external actor. Over half of risk statements implicated users as the primary threat actor. Moderators occurred ~20% of the time, with bots and external actors comprising the rest. Examples of each threat actor follow:

- **User**: “The captcha will remind the user that this is quite serious and will avoid spamming.” (Diaspora #4711)
Table 4. T&S risks identified in each repository. Taxonomy adapted from Thomas et al. [67] with additions in bold.

<table>
<thead>
<tr>
<th>Risk [67]</th>
<th>Description</th>
<th>Diaspora</th>
<th>Mastodon</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toxic Content</td>
<td>Content that users do not wish to see.</td>
<td>5</td>
<td>22</td>
<td>27</td>
</tr>
<tr>
<td>Content Leakage</td>
<td>Leak private content to wider audience.</td>
<td>19</td>
<td>5</td>
<td>24</td>
</tr>
<tr>
<td><strong>Undermoderation</strong></td>
<td>Moderation that is slow or ineffective.</td>
<td>6</td>
<td>11</td>
<td>17</td>
</tr>
<tr>
<td>Overloading</td>
<td>Force target to deal with a sudden influx of content.</td>
<td>6</td>
<td>11</td>
<td>17</td>
</tr>
<tr>
<td>Other</td>
<td>Risks that do not fit into any other category.</td>
<td>5</td>
<td>11</td>
<td>16</td>
</tr>
<tr>
<td>False reporting</td>
<td>Use of content reporting system with malintent.</td>
<td>6</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td><strong>Impersonation / Faulty Accounts</strong></td>
<td>Deceive others about identity.</td>
<td>5</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Lockout and Control</td>
<td>Interfere with access to a user’s account or any data.</td>
<td>3</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td><strong>Overmoderation</strong></td>
<td>Moderation that is too invasive or drastic.</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Surveillance</td>
<td>Aggregate or monitor user data.</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

- **Moderator:** “Moderators [can] access private [content]” (Mastodon #6986)
- **Bot:** “The current one is very bad at preventing bot registrations.” (Diaspora #8342)
- **External Actor:** “...risk of a hostile instance harvesting the private messages of unlocked users.” (Mastodon #4296)

We also carried out the risk identification step of the ISO risk management process [29]. Table 4 displays the risk taxonomy and frequencies across 137 risk statements. Toxic content is of particular interest in Mastodon, while Diaspora is most concerned with content leakage. Mastodon also saw more mentions of under moderation concerns rather than over moderation. These differences suggest that Mastodon is more focused on unwanted content on the platform and moderation resources to handle T&S risks. Meanwhile, Diaspora values data protection and respecting user privacy.

Examining risk statements over time, mentions of toxic content peaked from 2016-2019, which contained 24 statements — only 3 were from Diaspora. However, the content leakage risk saw an initial spike from Diaspora from 2011-2014, but another wave of activity began in 2016 with a crescendo in 2018 (roughly half of the activity went to each platform).

**Finding 2:** Users are the most common threat actors followed by moderators, external actors, and bots. Under moderation and overloading are secondary concerns for both platforms. Mastodon is primarily concerned with toxic content, while Diaspora focuses on content leakage. Content leakage was a concern 1–4 years after Diaspora was created, but a subsequent spike in activity occurred for both platforms in 2016-2018.

### 5.3 RQ3: Risk treatment: What treatment options are proposed in T&S issues? How are they selected?

To understand the risk treatment process, we identify treatment patterns, rationales of treatment selections, and assess the effectiveness of the process itself.

To study treatment patterns, we performed thematic coding on treatment options that treat T&S issues. We term these overarching themes as T&S Engineering patterns. The initial taxonomy was adopted from Koscik [38] and extended in this study (Table 5).

First, we consider the options that were proposed by discussion members. Table 5 displays each pattern and the frequencies. Add moderation is the most frequently proposed pattern, followed by require consent.

Based on the definitions of each pattern, we superimpose onto the previous context diagram (Figure 1) when each pattern intervenes and who each pattern relies on. The new context diagram is shown in Figure 4. We split the diagram based on proactive patterns that intervene before an interaction occurs and reactive patterns that intervene afterward.
Table 5. T&S Engineering Patterns. Patterns were coded from treatment option statements that treat T&S issues. Parenthesized digits are the total number of occurrences while un-parenthesized are the number of unique issues each pattern appears in. P/R indicates proactive vs. reactive patterns. Bold patterns are an addition to the taxonomy adopted from Koscik [38].

<table>
<thead>
<tr>
<th>Pattern Description</th>
<th>Example</th>
<th>P/R</th>
<th>Proposed</th>
<th>Chosen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add moderation</td>
<td>Add or improve moderation tools</td>
<td>&quot;User groups with ACL would be great though, so we could have multiple admins and/or a moderation team with access to reports.&quot; (Mastodon #811)</td>
<td>R</td>
<td>20 (35)</td>
</tr>
<tr>
<td>Require consent</td>
<td>Ask for approval from involved stakeholders</td>
<td>&quot;I think it should be unchecked, for privacy reasons.&quot; (Diaspora #4343)</td>
<td>P</td>
<td>15 (21)</td>
</tr>
<tr>
<td>Improve filters</td>
<td>Allow users to better control the content they see</td>
<td>&quot;I think the exploitation can be reduced arbitrarily to any personal preference by selecting from whom there can be invites.&quot; (Mastodon #7869)</td>
<td>R</td>
<td>7 (16)</td>
</tr>
<tr>
<td>Reduce visibility</td>
<td>Limit when a feature can be used</td>
<td>&quot;Users should not be allowed to invite users who have blocked or muted them.&quot; (Mastodon #7869)</td>
<td>P</td>
<td>8 (12)</td>
</tr>
<tr>
<td>Improve registration</td>
<td>Bolster user trustworthiness checks</td>
<td>&quot;About spam, what about a captcha during registration?&quot; (Diaspora #4616)</td>
<td>P</td>
<td>6 (8)</td>
</tr>
<tr>
<td>Reduce audience</td>
<td>Limit exposure of content</td>
<td>&quot;Or should their future participations in the conversation be hidden from the view of the person who has ignored them?&quot; (Diaspora #7612)</td>
<td>R</td>
<td>6 (7)</td>
</tr>
<tr>
<td>Interaction intervention</td>
<td>Intervene before users contact others</td>
<td>&quot;I think we should add a captcha when reporting a post.&quot; (Diaspora #4711)</td>
<td>P</td>
<td>3 (5)</td>
</tr>
<tr>
<td>Improve registration</td>
<td>Bolster user trustworthiness checks</td>
<td>&quot;In that case (to avoid harrassment) the tagged user should still be notified.&quot; (Mastodon #6469)</td>
<td>R</td>
<td>3 (4)</td>
</tr>
<tr>
<td>Reduce data</td>
<td>Remove unnecessary data from platform</td>
<td>&quot;Is it even possible not to generate OpenGraph info, if the post is marked as NSFW?&quot; (Diaspora #7962)</td>
<td>P</td>
<td>4 (4)</td>
</tr>
<tr>
<td>Reduce interaction</td>
<td>Limit how a feature can be used</td>
<td>&quot;Blocking someone should make it so that any of their replies to your posts should no longer be considered threaded.&quot; (Mastodon #4669)</td>
<td>P</td>
<td>2 (2)</td>
</tr>
<tr>
<td>Remove feature</td>
<td>Take out feature</td>
<td>--</td>
<td>P</td>
<td>0 (0)</td>
</tr>
</tbody>
</table>

Additionally, we signify in color the party that each pattern relies on. 7 of the identified patterns are **proactive** in nature, while 5 are **reactive**. 4 patterns rely on humans, but the other 8 are fully automated.

**Fig. 4. SMP context diagram with T&S Engineering patterns.** Patterns that intervene to the left of the dashed line are **proactive** and those to the right are **reactive**. See the legend for more detail.

By platform, Diaspora sees more **require consent** proposals along with **remove data** and **interaction intervention**. By contrast, Mastodon saw 16 **improve filters** suggestions compared to Diaspora’s zero, and more **moderation transparency** proposals. This comparison indicates that Diaspora is more focused on **reactive** patterns, while Mastodon is more concerned with **proactive** ones.
Table 6. T&S risk treatment rationales. We contextualize the taxonomy from Ko et al. to T&S [37]. Bold: new categories.

<table>
<thead>
<tr>
<th>Result</th>
<th>Rationale</th>
<th>Description</th>
<th>Example</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safety</td>
<td>Protects user from T&amp;S risks</td>
<td>&quot;The primary motivator would be that I just don’t want to see Bad Person’s bad posts while browsing in-app.&quot; (Mastodon #7741)</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>Moderator efficiency</td>
<td>Allows moderators to easily complete desired actions</td>
<td>&quot;Of course moderators can decide not to use direct messages, but moderation in the open is mostly not very productive.&quot; (Mastodon #8969)</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Feasibility</td>
<td>Ease of implementation</td>
<td>&quot;So, populating the checklist shouldn’t be hard.&quot; (Mastodon #823)</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Flexibility</td>
<td>Handles a variety of use cases</td>
<td>&quot;It could be a great compromise between letting users do all what they want and deleting their accounts once for all.&quot; (Diaspora #5564)</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Clarity</td>
<td>Provides clear experience to users</td>
<td>&quot;This is because in its current iteration, that is what it is, a ‘hard mute’.&quot; (Mastodon #231)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td>Prevents unwanted data access</td>
<td>&quot;It is an issue that admins can access unflagged private/direct messages.&quot; (Mastodon #6986)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>User efficiency</td>
<td>Allows user to easily complete desired action</td>
<td>&quot;Currently one needs to go to that particular post by clicking on the time stamp.&quot; (Diaspora #1667)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Annoyance</td>
<td>Removes unnecessary hindrance to user activity</td>
<td>&quot;It means a million extra clicks...to interact with the thread.&quot; (Mastodon #1123)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Unsafety</td>
<td>Adverse effect to user T&amp;S</td>
<td>&quot;On Twitter, DMs became a terrible spam vector and links in them were banned to try and mitigate this.&quot; (Mastodon #90)</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Infeasibility</td>
<td>Difficulty of implementation</td>
<td>&quot;I'm not aware of a technical possibility to prevent [unpermitted access] in a distributed network.&quot; (Diaspora #3863)</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Federation incompatibility</td>
<td>Not possible due to sharing protocols</td>
<td>&quot;This is technically very difficult to do right now in a federated manner, because we don’t support editing.&quot; (Diaspora #2123)</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Insecure</td>
<td>Susceptible to unwanted data access</td>
<td>&quot;This is to avoid accidental leak of a private post to an unwanted recipient and makes the federation protocol a lot easier as a side effect.&quot; (Diaspora #656)</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Inconsistency</td>
<td>Conflicts with design or user expectations</td>
<td>&quot;If it’s been read already, then it’s mutual property&quot; (Diaspora #1828)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Uncertainty</td>
<td>Unclear design or T&amp;S environment</td>
<td>&quot;That’s a good point, and we’ll probably revisit this in a week or so to see how people are using it.&quot; (Diaspora #3609)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Annoyance</td>
<td>Adds unnecessary hindrance to user activity</td>
<td>&quot;Every user has to subscribe to the shared blocklist.&quot; (Mastodon #1092)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Unclarity</td>
<td>Complicated or convoluted user experience</td>
<td>&quot;Mastodon aims to be usable by ‘non-tech-savvy’ people (I guess that implies basic ‘online safety’ measures as well)&quot; (Mastodon #8540)</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Finding 3: Add moderation is the most commonly proposed pattern followed by require consent and improve filters. The majority of discovered patterns are proactive in nature. Diaspora sees more of these proposals than Mastodon, which had a heavy emphasis on the improve filters pattern.

Next, we analyze what patterns are actually chosen by engineers and why. Figure 4 shows that proactive patterns are chosen less frequently by engineers and chosen options rely on users or moderators more often than not. Table 5 indicates that improve registration had the highest acceptance rate (38%) and add moderation had the lowest (23%). Table 6 compares the most common reasons for acceptance (merged) or rejection (no action) of a proposal.

Among the set of proposed treatments, they tend to be proactive (not reactive) and automated (not relying on humans). However, most chosen options are reactive and rely on human intervention. Moderator efficiency was cited in many accepted proposals (e.g., supporting human intervention), while federation incompatibility was a common reason to take no action on an opened issue (e.g., preventing automation).

Finding 4: Reactive patterns are chosen 13/22 times and those that involve humans are chosen 17/22 times. Most accepted treatments were associated with safety and moderation efficiency. Rejected treatments were commonly unsafe for users, infeasible, or exhibited federation incompatibility.

Last, we consider T&S issue status and age to get a sense of how effective the T&S risk treatment process is. We found that more than one-third of T&S issues are still open with no resolution (Figure 5). Closed T&S issues took almost 5 months longer to resolve than the average issue closure time. Figure 5 also shows that Diaspora has closed issues with no action more frequently than Mastodon.
Finding 5: T&S issues take 147 days longer to resolve compared to the average closure time. 38% of identified T&S issues remain open. Diaspora is less prone (53%) than Mastodon (87%) to making platform changes in response to T&S issues (87%) (Figure 5).

6 DISCUSSION AND FUTURE WORK

6.1 Recommendations for OSS SMPs

We suggest several ways in which OSS SMPs might improve their T&S risk management process.

6.1.1 Document risk sources and treatment options. In the OSS SMPs we studied, knowledge of risky features, risk factors, and risk treatments is distributed across project personnel and documents (e.g., distinct issues). We identified patterns within these features (Table 3), factors (Table 4), treatments (Table 5), and rationales (Table 6). Patterns can accelerate the engineering process: prior conversations and decisions could be tracked to guide future T&S discussions. This would promote consistency in decision-making and let precedent resolve dispute.

6.1.2 Explore proactive solutions. In our data, we examined 60 issues of which 19 were resolved with a change. As illustrated in Figure 4, most of these solution approaches were reactive rather than proactive. They generally shared risk between users and moderators of the system. Would an ounce of prevention be better than a pound of cure? Can SMPs pursue proactive patterns instead to prevent T&S risks before they are realized?

6.1.3 Stay vigilant. Our study of the T&S defect arrival rate (Figure 3) showed that T&S issues manifest later than other defects, and remain present throughout SMP lifespans. As a non-functional requirement similar to cybersecurity, T&S will likely remain a concern for the lifetime of the project.
6.2 Future Work

Our exploratory research identified several research opportunities to improve T&S Engineering.

6.2.1 T&S Engineering Pattern Catalog. From findings 2 and 3, there are clear problem and solution themes within SMPs. Tables 4 and 5 provide the first empirically grounded patterns for T&S problems and solutions in SMPs. Further work in taxonomization, e.g., expanding to more issues or other OSS SMPs (Table 1), could improve this catalog. The T&S risks on commercial SMPs could also be incorporated, e.g., following the method of Anandayuvoraj & Davis [4], although the solution patterns are sometimes opaque. Figure 4 offers a starting point for organizing such work.

The merits of such a catalog must also be assessed. Context dictates which pattern, if any, may be suitable. We conjecture that T&S risks recur frequently enough within and across SMPs that a pattern catalog would simplify the selection and treatment of T&S risk, resulting in more consistent decisions made more quickly.

6.2.2 Improved T&S Testing. Surprisingly, in the T&S discussions we studied, testing was never mentioned. Operationalizing T&S for automated testing is an open challenge. However, due to the contextual nature of T&S risks, fully automated techniques such as [5, 27] could be limited. For example, automated T&S testing could check that basic user boundaries are respected, but this would require models for normal and abnormal user behavior, for user boundaries, for consent, and so on. A possible starting point is the usability testing literature [10, 57].

6.2.3 Automated Content Moderation in OSS SMPs. Commercial SMPs rely heavily on automated moderation, while OSS SMPs tend to use human moderation. Human moderation has limits — under moderation is a frequent T&S risk in OSS SMPs (Table 4). However, developing accurate automated moderation has proven challenging because of the amount of contextual information required to make a judgment. To what extent can automated content moderation be incorporated into OSS SMPs? Is a decentralized OSS SMP instance easier to moderate (e.g., a more homogeneous user base) than a centralized SMP? Investigating automated content moderation could strengthen this weak point in the T&S risk environment. However, there are many T&S considerations to such a proposal, including: whether and how moderators/users can opt in to this feature; ensuring that data is handled properly; and communicating any other residual risks to involved stakeholders. Furthermore, OSS SMP stakeholders may be unwilling to adopt automated content moderation due to highly-publicized failures in commercial SMPs. Understanding these human factors and the interplay between commercial and OSS SMPs could advance the conversation.

6.2.4 T&S Improvements in Federated Protocols. Federation incompatibility was cited in 7 proposal rejections (Table 6). Thus, federation protocols expose OSS SMPs to substantial risk. Adding safety features within the protocol (e.g., anti-spam measures [63]) could increase the feasibility of some T&S treatments on SMPs. End-to-end arguments in system design suggest limits to the T&S impact of a protocol [56], but perhaps some improvement is possible.

6.2.5 T&S By Design. As discovered in finding 4 and discussed in §6.1.2, many of the T&S engineering patterns we observed were reactive, addressing T&S issues by intercepting problematic behavior or content after it has been generated. Prior works have studied how T&S can be incorporated into comment thread design [59] and SMP design [36], but as yet there is no general agenda for T&S by Design. This direction should be informed by fields such as Privacy by Design [14, 39] and Security by Design [13, 14]. Rubinstein & Good argue that past SMP privacy failures could be avoided through a design approach [54]. Leveraging this work to inform T&S engineering processes may allow engineers to move from re-actively improving T&S to proactively promoting T&S by design.
7 THREATS TO VALIDITY

Internal validity. Our methodological choices that could affect our findings. First, our work relied on qualitative analysis. To reduce bias, we measured inter-rater agreement. To promote comparisons across studies, we used existing taxonomies, extending them as needed. Second, our work mined GitHub. This carries concomitant general concerns [6, 31]. There is also a Diaspora-specific concern. Diaspora uses a separate forum to discuss preliminary feature proposals [21]. Some of these proposals are subsequently filed on GitHub; we only studied such. This data source was omitted because those proposals do not include actions taken by OSS engineers.

External validity. The primary threat to this work is its generalizability. We examined two open-source SMPs with decentralized architectures, omitting other open-source SMPs and all commercial SMPs (which have different goals for their platforms, centralized architectures, and greater resources). We note two mitigating features of our work. First, although the SMPs we studied are a fraction of the size of SMPs such as Facebook, they nevertheless have over 8 million users — T&S concerns affecting 8 million users are worth studying. Second, although we studied open-source decentralized SMPs, we built our analysis on top of existing taxonomies derived from commercial SMPs. Our data fit these taxonomies, suggesting similarities between the contexts, although in each case we observed new behaviors that required extending the taxonomies.

As a secondary concern, we studied only N=60 issues, 30 from each SMP. A larger sample size could increase the scope of our findings. We note that we analyzed 73% of Diaspora issues (Table 2), indicating that the data was approaching exhaustion for that project. Furthermore, even within this sample, we were able to extend each existing taxonomy that we applied.

Construct validity. There is no precise definition of “Trust & Safety”. Since T&S is fundamentally a contextual and personal construct, others might reach different conclusions from our data. We operationalized T&S in the terms used by T&S researchers and T&S practitioners such as TSPA, and used those terms to retrieve relevant issues on GitHub. We then analyzed those issues using our own understanding of T&S risks (§2) by leveraging an ISO risk management standard [29]. However, there is no guarantee that the OSS engineers were using the same terminology. We mitigated this by measuring information retrieval on our keywords.

8 CONCLUSION

Promoting Trust & Safety (T&S) on SMPs is a major challenge that involves users, moderators, policymakers, and regulators. Software engineering matters too: through design, implementation, and validation, software engineers can reduce an SMP’s T&S risks.

We conducted the first empirical study of T&S risks on SMPs from a software engineering perspective. We studied 60 T&S-related GitHub Issues for the two most popular open-source SMPs, Mastodon and Diaspora. Our work identified novel SMP risks, engineering patterns, and resolution rationales. Our key findings are: (1) T&S issues persist throughout a platform’s lifetime and mostly require design changes; (2) T&S issues are hard to resolve or remain open; (3) Selected treatments are mostly reactive rather than proactive; and (4) Selected treatments mostly share risk with users or moderators, despite many alternatives. Our work suggests that, in open-source SMPs, there is currently no systematic engineering approach to promoting T&S. We show opportunities for research on software design, decision-making, and validation for T&S in SMPs.
9 DATA AVAILABILITY

Replication data is available on request, including codebook, sampled issues, models, and multi-rater codes.

ACKNOWLEDGMENTS

We thank A. Kazerouni, A. Marwick, A. Quinn, A. Tewari, and T. Zhang for their input.

REFERENCES
[23] Lexi Galantino. 2019. Trust & Safety Engineering @ GitHub. https://www.youtube.com/watch?v=UC3Y9e0xjFQ
An Exploratory Empirical Study of Trust & Safety Engineering in Open-Source Social Media Platforms


