Harvard Journal of Law & Technology
Volume 39, Number 1 Fall 2025

TAINTED SOURCE CODE

*Bryan H. Choi**

ABSTRACT

Open-source software has long eluded tort liability. Fierce ideological commitments and sticky license terms support a long tradition of forbearance against penalizing harmful or negligent work in open-source communities. The free, noncommercial, distributed, and anonymous characteristics of open-source contributions present additional obstacles to legal enforcement.

The exponential rise in software supply chain attacks has given new urgency to the problem of bad open-source code. Yet, current approaches are unlikely to meaningfully improve open-source security and safety. On the one hand, technological tools and self-governance mechanisms remain woefully underdeveloped and underutilized. On the other hand, liability proposals that place all the burden on commercial vendors to inspect the open-source packages they use are impractical solutions that ignore how software is built and maintained.

This Article argues that donated code should be subject to tort liability by analogy to the law of tainted food and blood donations. Food safety law is the progenitor of modern tort law, and it reveals an older set of tensions between altruistic efforts to address societal hunger and the need for accountability in regulating the quality of food supply chains. At common law, the charitable nature of a donation is a nonfactor in determining liability. Legislatures have intervened to provide safe harbors, but only up to an extent. This nuanced history offers a principled path forward for extending a liability framework to donations of open-source code.

---

* Associate Professor of Law, University of Colorado Law School. I thank Derek Bambauer, Aeva Black, Jack Cable, David Clark, Brian Eschels, Laura Heymann, Gus Hurwitz, Amy Landers, Michael Madison, Paul Ohm, Guy Rub, Chinmayi Sharma, Bryant Walker Smith, Margo Seltzer, Rebecca Wexler, Christopher Yoo, and participants of the Tenth Annual Computer Science and Law Roundtable and the Law and Technology Workshop for helpful comments at early stages of this project. I also thank Rebecca Cook, Kevin Gibbons, Damini Mohan, and Rama Naboulsi for invaluable research assistance. All opinions, errors, and omissions are my own. This work was supported in part by NSF CCF-2131531.

TABLE OF CONTENTS

I. INTRODUCTION

Software liability has an open-source problem. Audits and studies show that open-source code pervades up to ninety-six percent of commercial and government software, and that up to seventy-four percent of those codebases contain *high-risk* vulnerabilities.[1] Yet, the majority

1. *See* SYNOPSYS, 2024 OPEN SOURCE SECURITY AND RISK ANALYSIS REPORT 6 (2024), https://static.carahsoft.com/concrete/files/1617/1597/8665/2024_Open_Source_Security_and_Risk_Analysis_Report_WRAPPED.pdf [https://perma.cc/PM5N-T8TU] ("Of the 1,067 codebases analyzed by the Black Duck Audit Services team and used as the base data for this year's OSSRA report, 96% contained open source. Seventy-seven percent of all the source code and files scanned originated from open source code . . . . Seventy-four percent of those codebases contained high-risk vulnerabilities, a significant increase from 2022, when only 48% of the codebases were found to contain high-risk vulnerabilities."); SONATYPE, 9TH ANNUAL STATE OF THE SOFTWARE SUPPLY CHAIN 6 (2023), https://www.sonatype.com/hubfs/SSC/2023%20Sonatype-%209th%20Annual%20State%20of%20the%20Software%20Supply%20Chain-%20Update.pdf [https://perma.cc/6N39-UCZA] (estimating that "up to 90% of the code we run in production is of open source origin"); Request for Information on Open-Source Software Security: Areas of Long-Term Focus and Prioritization, 88 Fed. Reg. 54315 (Aug. 10, 2023); The White House, *Readout of White House Meeting on Software Security*, THE WHITE HOUSE (Jan. 13, 2022), https://bidenwhitehouse.archives.gov/briefing-room/statements-releases/2022/01/13/readout

of open-source projects are poorly maintained or abandoned, resulting in broad swaths of dilapidated code and easy targets for malicious attackers.[2] Not surprisingly, the number of attacks via open-source software supply chains has increased at an exponential rate since 2021,[3] after the widely publicized exposures of severe vulnerabilities such as Log4Shell and supply-chain attacks such as SolarWinds and NotPetya.[4] Despite these threats, known vulnerabilities that originate in open-source projects continue to remain unpatched many years after discovery.[5]

Puzzlingly, despite the known severity of open-source risk, tort liability for open-source contributions is viewed with trepidation. A recent meta-analysis of legal scholarship found that *none* of the reviewed articles "endorsed the idea of including developers of [open-source software] in a software liability regime."[6] National cybersecurity

---

-of-white-house-meeting-on-software-security/ [https://perma.cc/EL2E-3J7Z] ("Most major software packages include open source software . . . . [which] brings unique value, and has unique security challenges.").

2. *See* SONATYPE 2023 REPORT, *supra* note 1, at 19 (finding only eleven percent of open-source projects are actively maintained).

3. *See id.* at 11 (reporting "twice as many supply chain attacks as the cumulative numbers in previous years"); SONATYPE, 2021 STATE OF THE SOFTWARE SUPPLY CHAIN REPORT 10 (2021), https://www.sonatype.com/hubfs/Q3%202021-State%20of%20the%20Software%20Supply%20Chain-Report/SSSC-Report-2021_0913_PM_2.pdf [https://perma.cc/542X-8ZGM] (finding a 650 percent increase in attacks on open-source software vendors).

4. *See* Chinmayi Sharma, *Tragedy of the Digital Commons*, 101 N.C. L. REV. 1129, 1132 (2023) (explaining the Log4Shell vulnerability and noting that "nearly *half* of global corporate networks experienced a successful exploit in the first five days following the vulnerability's discovery"); Dina Temple-Raston, *A 'Worst Nightmare' Cyberattack: The Untold Story of the SolarWinds Hack*, NPR (Apr. 16, 2021, at 10:05 ET), https://www.npr.org/2021/04/16/985439655/a-worst-nightmare-cyberattack-the-untold-story-of-the-solarwinds-hack [https://perma.cc/9PPS-M3BZ]; Andy Greenberg, *The Untold Story of NotPetya, the Most Devastating Cyberattack in History*, WIRED (Aug. 22, 2018, at 5:00 ET), https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/ [https://perma.cc/28FH-KMCL] (explaining that NotPetya was pushed out through a backdoor created by "routine updates" to accounting software called M.E.Doc).

5. *See* SYNOPSYS, *supra* note 1, at 4 (reporting that fourteen percent of codebases contain vulnerabilities older than ten years); *2023 Top Routinely Exploited Vulnerabilities*, Cybersecurity & Infrastructure Security Agency (CISA) (Nov. 12, 2024), https://www.cisa.gov/news-events/cybersecurity-advisories/aa24-317a [https://perma.cc/FW2V-XE3H] (listing Log4Shell as one of the top fifteen routinely exploited vulnerabilities more than two years after discovery).

6. MAIA HAMIN, SARA ANN BRACKETT & TREY HERR, DESIGN QUESTIONS IN THE SOFTWARE LIABILITY DEBATE 13–15 (2024), https://dfrlab.org/wp-content/uploads/sites/3/2024/01/AC_CSI_Liability_Design_Questions.pdf [https://perma.cc/M8XS-DGQZ]. *But see* TREY HERR, JUNE LEE, WILLIAM LOOMIS & STEWART SCOTT, ATLANTIC COUNCIL, BREAKING TRUST: SHADES OF CRISIS ACROSS AN INSECURE SOFTWARE SUPPLY CHAIN 29 (2020), https://www.atlanticcouncil.org/wp-content/uploads/2020/07/Breaking-trust-Shades-of-crisis-across-an-insecure-software-supply-chain.pdf [https://perma.cc/643A-9H55] ("The US Congress should extend final goods assembler liability to operators of major open-source repositories, container managers, and app stores."); John Speed Meyers & Paul Gilbert, *Questioning the Conventional Wisdom on Liability and Open Source Software*,

concerns have compelled the White House and leading lawmakers to recognize that software vendors need to be held legally accountable for releasing bad code.[7] Yet, even among software liability hawks who favor liability rules for commercial software, open-source software poses an uneasy quandary.[8] Conventional wisdom holds that because open source is a "public good," open-source contributions must be unfettered by legal liability as a matter of public policy.[9] Deterring open-source participation through punitive measures may seem to some like one step forward, two steps back. But open source, like any other software, is susceptible to careless errors. Meanwhile, the increasingly central role of open-source code in critical systems is making it harder to exonerate those errors.[10] Given these competing commitments, it is not immediately obvious from a cheapest-cost-avoider perspective why open-source ought to be categorically exempt from legal scrutiny.

The XZ Utils code poisoning attack is yet another recent illustration of how open source can taint the software supply chain.[11] XZ Utils

---

LAWFARE (Apr. 18, 2024, at 13:00 ET), https://www.lawfaremedia.org/article/questioning-the-conventional-wisdom-on-liability-and-open-source-software [https://perma.cc/9C3J-H8V5]; *see also* Request for Information on Open-Source Software Security, *supra* note 1, at 54316 (listing as one potential area of focus: "Applications of cybersecurity insurance and appropriately-tailored software liability as mechanisms to incentivize secure software development and operational environment practices").

7. *See* THE WHITE HOUSE, NATIONAL CYBERSECURITY STRATEGY 20 (2023), https://bidenwhitehouse.archives.gov/wp-content/uploads/2023/03/National-Cybersecurity-Strategy-2023.pdf [https://perma.cc/Z24X-345S] ("We must begin to shift liability onto those entities that fail to take reasonable precautions to secure their software . . . ."); U.S. CYBERSPACE SOLARIUM COMM'N, FINAL REPORT 76 (2020), https://cybersolarium.org/wp-content/uploads/2022/05/CSC-Final-Report.pdf [https://perma.cc/89ZQ-XN3P] (recommending that "Congress should pass a law establishing that final goods assemblers of software, hardware, and firmware are liable for damages from incidents that exploit known and unpatched vulnerabilities").

8. *See* THE WHITE HOUSE, *supra* note 7, at 21 (declaring that responsibility must be placed "not on . . . the open-source developer of a component that is integrated into a commercial product").

9. *See, e.g.*, Directive 2024/2853 of the European Parliament and of the Council of 23 October 2024 on Liability for Defective Products and Repealing Council Directive 85/374/EEC, recital 14, 2024 O.J. (L) 1 [hereinafter EU Product Liability Directive] ("In order not to hamper innovation or research, this Directive should not apply to free and open-source software developed or supplied outside the course of a commercial activity, since products so developed or supplied are by definition not placed on the market."); THE WHITE HOUSE, SECURING THE OPEN-SOURCE SOFTWARE ECOSYSTEM 4 (2024), https://bidenwhitehouse.archives.gov/wp-content/uploads/2024/01/Securing-the-Open-Source-Software-Ecosystem-OS3I-End-of-Year-Report-MASTERCOPY.pdf [https://perma.cc/Z9D6-EFEQ] ("Given that open-source software is a public good, ensuring open-source software's resilience is a technical necessity and a strategic imperative for protecting and promoting U.S. interests.").

10. *See* Sharma, *supra* note 4, at 1147 ("What makes open source valuable also makes it a unique risk . . . . When the same piece of code is used by hundreds of thousands of networks internationally, then one vulnerability in one project can take countless critical systems offline.").

11. *See* Dan Goodin, *The XZ Backdoor: Everything You Need to Know*, WIRED (Apr. 2, 2024, at 04:00 ET), https://www.wired.com/story/xz-backdoor-everything-you-need-to-

is a widely used data compression tool that is included in almost all Linux and Unix installations — hundreds of millions of computers. Despite its importance, the XZ project was an "unpaid hobby project" maintained by only one person, Lasse Collin, who had a history of going dormant for long stretches.[12] From 2021 to 2024, a group of unknown pseudonymous users carried out a stealth campaign to take over control of the XZ project. Beginning in 2021, a user going by "JiaT75" became increasingly active, building trust and credibility as an established contributor.[13] Then in 2022, a group of other pseudonymous users — "Jigar Kumar" and "Dennis Ens" — coordinated pressure on Collin to transfer control of the project. Collin conceded in January 2023 and gave commit access to JiaT75. More than a year later, in February 2024, JiaT75 issued a routine code update with a hidden backdoor exploit that was scheduled to be rolled out to major distributors such as Debian and Red Hat. Only luck and the keen eye of another unpaid hobbyist, Andres Freund, prevented the backdoor from being widely deployed and causing harm.[14]

Software supply chain failures are based on a simple premise: software is updated often.[15] Those updates inject new code — often

know/ [https://perma.cc/GPS9-T8PZ]; Bruce Schneier, *Backdoor in XZ Utils that Almost Happened*, LAWFARE (Apr. 9, 2024, at 15:30 ET), https://www.lawfaremedia.org/article/backdoor-in-xz-utils-that-almost-happened [https://perma.cc/L4XZ-76R4].

12. *See* Email from Lasse Collin, Re: [xz-devel] XZ for Java (June 8, 2022, at 03:28 ET), https://www.mail-archive.com/xz-devel@tukaani.org/msg00567.html [https://perma.cc/663N-9GTA]. The problem of burnout is commonly reported. *See, e.g.*, Nolan Lawson, *What It Feels Like to Be an Open-source Maintainer*, READ THE TEA LEAVES, (Mar. 5, 2017), https://nolanlawson.com/2017/03/05/what-it-feels-like-to-be-an-open-source-maintainer/ [https://perma.cc/7PQ8-DN8S]; Courtney Miller, Sophie Cohen, Daniel Klug, Bogdan Vasilesdcu & Christian Kaustner, *"Did You Miss My Comment or What?": Understanding Toxicity in Open Source Discussions*, 44 INT'L CONF. ON SOFTWARE ENG'G 710 (2022).

13. Although JiaT75 went by an East Asian name, "Jia Tan," security researchers suspect the account was an invented persona controlled by a hacker group based in an Eastern European or Middle Eastern time zone. *See* Andy Greenberg & Matt Burgess, *The Mystery of 'Jia Tan,' the XZ Backdoor Mastermind*, WIRED (Apr. 3, 2024, at 09:54 ET), https://www.wired.com/story/jia-tan-xz-backdoor/ [https://perma.cc/TC2Z-YALN].

14. *See* Schneier, *supra* note 11 (assessing that the XZ backdoor would have been "orders of magnitude more damaging" than the 2020 SolarWinds attack that affected 14,000 networks).

15. *See* NAT'L INST. OF STANDARDS & TECH. (NIST), DEFENDING AGAINST SOFTWARE SUPPLY CHAIN ATTACKS 4–5 (2021), https://www.cisa.gov/sites/default/files/publications/defending_against_software_supply_chain_attacks_508.pdf [https://perma.cc/3B3M-UMEK] (describing the three most common attack techniques as: (1) hijacking updates through infiltration, (2) hijacking updates by impersonating a trusted vendor, and (3) compromising open-source code libraries); HERR ET AL., *supra* note 6, at 10 ("A software supply chain attack occurs when an attacker accesses and modifies software in the complex software development supply chain to compromise a target farther down on the chain by inserting their own malicious code.").

automatically — into an existing software system.[16] To be sure, proprietary "closed source" systems can be poisoned by tainted code, whether through routine software maintenance or through sophisticated takeover attacks.[17] But the open-source development model presents additional risk pathways for bad code to contaminate the software supply chain. Updates are crowdsourced through many pseudonymous volunteers with dubious credentials.[18] Most projects are critically undermanned and under-resourced.[19] Projects are often abandoned.[20] The adoption and provenance of open-source code is poorly tracked[21] and often results in massive numbers of hidden dependencies.[22]

---

16. *See* ENDURING SEC. FRAMEWORK, SECURING THE SOFTWARE SUPPLY CHAIN: RECOMMENDED PRACTICES FOR CUSTOMERS 1 (2022), https://www.cisa.gov/sites/default/files/2024-08/SECURING_THE_SOFTWARE_SUPPLY_CHAIN_SUPPLIERS_508.pdf [https://perma.cc/4AS3-VDDY] (explaining that "threat actors proactively inject malicious code into products that are then legitimately distributed downstream through the global software supply chain").

17. *See, e.g.*, Brian Krebs, *Global Microsoft Meltdown Tied to Bad Crowdstrike Update*, KREBS ON SEC. (July 19, 2024), https://krebsonsecurity.com/2024/07/global-microsoft-meltdown-tied-to-bad-crowstrike-update/ [https://perma.cc/B54P-ZH79].

18. *See* SONATYPE, 2020 STATE OF THE SOFTWARE SUPPLY CHAIN REPORT 6 (2020), https://www.sonatype.com/hubfs/SSC/SON_SSSC-Report-2020_sept23.pdf [https://perma.cc/9UGH-H2NN] ("Open source projects rely on contributions from thousands of volunteer developers, and discriminating between community members with good or malicious intent is difficult, if not impossible."); HERR ET AL., *supra* note 6, at 5 ("The security of open-source projects, and the apparent ease with which attackers can introduce insecure code, is a continuing concern. The fluidity with which anyone can commit code to an open-source project is at once a core strength and glaring weakness.").

19. *See* NADIA EGHBAL, WORKING IN PUBLIC 8 (2020); Jessy Ayala, Yu-Jye Tung & Joshua Garcia, *A Mixed-Methods Study of Open-Source Software Maintainers on Vulnerability Management and Platform Security Features*, 34 USENIX SEC. SYMP. 2105, 2113 (2025) (noting that "large-scale software ecosystems like Python Package Index and NPM are primarily made up of projects with a single maintainer"); Josh Bressers, *Open Source Is One Person*, OPEN SOURCE SEC. (Aug. 28, 2025), https://opensourcesecurity.io/2025/08-oss-one-person/ [https://perma.cc/G6EJ-DB4Y] (explaining that about 7 million of 11.8 million open-source projects are maintained by one person); *see also* Brian Krebs, *Microsoft Patch Tuesday, December 2021 Edition*, KREBS ON SEC. (Dec. 14, 2021), https://krebsonsecurity.com/2021/12/microsoft-patch-tuesday-december-2021-edition/ [https://perma.cc/EY63-PVJV] (describing Log4Shell as "a 90s style Java vulnerability in an open source module, written by two volunteers with no funding, used by large cybersecurity vendors").

20. *See* SYNOPSYS, *supra* note 1, at 4 (finding forty-nine percent of assessed codebases "had components that had no development activity in the past 24 months").

21. *See* Sharma, *supra* note 4, at 1146 ("Neither maintainers nor users of a project know where the code is and what it is being used for, which precludes privately notifying other affected parties of an issue, distributing a fix, collaborating on development, or preventing use of an insecure project version.").

22. *See* SONATYPE 2020 REPORT, *supra* note 18, at 6 ("Open source projects themselves typically incorporate hundreds — if not thousands — of dependencies from other open source projects . . . . The sheer volume of open source in use and the massive number of dependencies makes it difficult to quickly evaluate the quality and security of every new version of a dependency."); SONATYPE, 2016 STATE OF THE SOFTWARE SUPPLY CHAIN REPORT 8 (2016), https://www.sonatype.com/hubfs/SSC/2016_State_of_the_Software_Supply_Chain_Report.pdf [https://perma.cc/2AZZ-TGVV] (observing that "modern [software] applications

In seeking to harden the software supply chain, the White House has focused primarily on commercial software vendors while deferring discussion of open-source software. At least three major focuses have emerged from those discussions. First is the threshold question whether software harms should be governed primarily by contract law or by tort law.[23] Since the mid-1990s, courts have deferred mainly to contract law, particularly where there is only economic loss without physical injury.[24] Doing so has meant there is little or no remedy for software harms, because software licenses stipulate broad limitations of liability and waivers of warranty.[25] Consequently, more recent scholarship has begun to attack such provisions as overbroad and against public policy,[26] or to argue that public policy should be amended to invalidate

typically consist of 80%–90% component parts," which themselves may encapsulate open-source components).

23. *See* THE WHITE HOUSE, *supra* note 7, at 21 (pledging support for legislation that would "prevent manufacturers and software publishers with market power from fully disclaiming liability by contract"); Jim Dempsey, *Standards for Software Liability: Focus on the Product for Liability, Focus on the Process for Safe Harbor*, LAWFARE (Jan. 23, 2024, at 11:15 ET), https://www.lawfaremedia.org/article/standards-for-software-liability-focus-on-the-product-for-liability-focus-on-the-process-for-safe-harbor [https://perma.cc/M4A7-V6RG]; *see also* EU Product Liability Directive, *supra* note 9, at recital 13 ("[I]t should be clarified in this Directive that software is a product for the purposes of applying no-fault liability.").

24. The conventional wisdom is set forth by Michael D. Scott, *Tort Liability for Vendors of Insecure Software: Has the Time Finally Come*, 67 MD. L. REV. 425, 437–41, 457, 471 (2008) (explaining that warranty disclaimers and liability limitation clauses are generally upheld unless they are found to be unconscionable, and that contractual limitations on liability generally displace tort liability unless there is physical, noneconomic loss); Michael L. Rustad & Thomas H. Koenig, *The Tort of Negligent Enablement of Cybercrime*, 20 BERKELEY TECH. L.J. 1553, 1565 (2005) (explaining that "courts' attitudes have [changed since the mid-1990s] in favor of broad enforceability of mass market license agreements"). *But see* Charlotte A. Tschider, *Unto the (Data) Breach*, 59 U. RICH. L. REV. 591, 628 (2025) (noting that many claims relating to data breaches allege torts); Frances E. Zoller, Andrew McMullin, Sandra N. Hurd & Peter Shears, *No More Soft Landings for Software*, 21 SANTA CLARA COMPUT. & HIGH TECH. L.J. 745, 764–65 (2005) (explaining that courts have been "inconsistent" and have "oftentimes strained" to find ways to bypass limitations and disclaimers "to find a remedy that a strict reading of the contract would not seem to permit"); Mark A. Lemley, *Terms of Use*, 91 MINN. L. REV. 459, 462–63 (2006) ("[T]hose courts that have enforced shrinkwrap and clickwrap licenses against consumers have protected consumers against certain clauses considered unreasonable."); Catherine M. Sharkey, *Can Data Breach Claims Survive the Economic Loss Rule?*, 66 DEPAUL L. REV. 339, 378, 380 (2017) (observing that courts are finding tort liability is warranted for economic losses in data breach cases, and arguing that these decisions can be justified as "a stand-in for public regulation").

25. *See* Jane Chong, *Bad Code: The Whole Series*, LAWFARE (Nov. 4, 2013, at 12:46 ET), https://www.lawfaremedia.org/article/bad-code-whole-series [https://perma.cc/GXJ4-KJUX]; *see also* Bryan H. Choi, *Crashworthy Code*, 94 WASH. L. REV. 39, 68 (2019) (describing other doctrines courts have used to avoid deciding software liability cases).

26. *See* Nancy S. Kim, *Adhesive Terms and Reasonable Notice*, 53 SETON HALL L. REV. 85, 127 (2022) ("Courts tend to determine that an exculpatory agreement violates public policy if there is a lack of bargaining power."); Rustad & Koenig, *supra* note 24, at 1562 ("Mass market license agreements are classic examples of adhesion contracts in which the licensor routinely disclaims all meaningful warranties and remedies, and the manufacturer reallocates the risk of loss to the user community for all failures of performance."); *see also* Andrew A. Schwartz, *Consumer Contract Exchanges and the Problem of Adhesion*, 28 YALE J. ON REG. 313, 347 (2011).

such provisions.[27] Secondly, if tort law applies, what should be the standard of care? The commentary has divided between those who believe a standard of "reasonableness" can be meaningfully articulated, and those who worry it cannot.[28] The latter camp divides further: some have suggested that enforcement should focus on a minimum floor of obviously unacceptable conduct, while others have argued that there is too much heterogeneity in software practices to locate even this first incremental step.[29] Third, policymakers have asked whether a safe harbor can be crafted to shield software developers who comply with designated best practices.[30] Here, too, consensus is unsettled on whether a safe harbor can be constructed without abdicating meaningful accountability for shielded entities.[31]

Add a fourth challenge to that mix: how should software liability extend to open-source components? Much of the existing discussion has focused on commercial, for-profit entities. But what makes open-source code distinctive is the free and charitable nature of the contributions. What is the law of the gift horse?

Part II opens with an overview of current approaches to software supply chain security. Overwhelmingly, the recommendations and

---

27. *See* Michael L. Rustad, *Cancel Carte Blanche for the Information Industries: Federalizing U.C.C. Article 2*, 89 MO. L. REV. 59, 152 (2024) (advocating federal U.C.C. reforms such that software developers would "no longer be able to disclaim warranties, limit liability, or assert a privity defense in a case involving software"); Robert A. Hillman, *Contract Law in Context: The Case of Software Contracts*, 45 WAKE FOREST L. REV. 669, 676 (2010) ("The ALI Principles thus include a nondisclaimable warranty of no hidden material defects of which the transferor is aware.").

28. *Compare* Chinmayi Sharma & Benjamin C. Zipursky, *Who's Afraid of Products Liability? Cybersecurity and the Defect Model*, LAWFARE (Oct. 19, 2023, at 10:24 ET), https://www.lawfaremedia.org/article/who-s-afraid-of-products-liability-cybersecurity-and-the-defect-model [https://perma.cc/25EP-PRWV], Catherine Sharkey, *A Products Liability Framework for AI*, 25 COLUM. SCI. & TECH. L. REV. 240, 249 (2024), *and* Asaf Lubin, *On Software Bugs and Legal Bugs: Product Liability in the Age of Code*, 100 IND. L.J. 1891, 1896–97 (2025), *with* Dempsey, *supra* note 23, Derek Bambauer, *Cybersecurity for Idiots*, 106 MINN. L. REV. HEADNOTES 172, 195 (2021), *and* Bryan H. Choi, *Software as a Profession*, 33 HARV. J.L. & TECH. 557, 559 (2020).

29. *Compare* Derek E. Bambauer & Melanie J. Teplinsky, *Shields Up for Software*, LAWFARE (Dec. 19, 2023, 2:07 PM), https://www.lawfaremedia.org/article/shields-up-for-software [https://perma.cc/AF54-KW87], *and* Choi, *supra* note 25, *with* Steve Lipner, *Incentives for Improving Software Security: Product Liability and Alternatives*, LAWFARE (May 14, 2024, at 12:34 ET), https://www.lawfaremedia.org/article/incentives-for-improving-software-security-product-liability-and-alternatives [https://perma.cc/9YX9-4D95].

30. *See* Derek E. Bambauer & Melanie J. Teplinsky, *Standards of Care and Safe Harbors in Software Liability: A Primer*, LAWFARE (May 31, 2024, at 09:39 ET), https://www.lawfaremedia.org/article/standards-of-care-and-safe-harbors-in-software-liability--a-primer [https://perma.cc/2HDG-FLE7] (noting that a significant challenge in designing a software liability regime is to avoid "crippling the software industry" or "caus[ing] innovators to abandon the market"); Charlotte A. Tschider, *Will a Cybersecurity Safe Harbor Raise All Boats?*, LAWFARE (Mar. 20, 2024, at 09:42 ET), https://www.lawfaremedia.org/article/will-a-cybersecurity-safe-harbor-raise-all-boats [https://perma.cc/GG3V-66C3].

31. *Compare* Bambauer & Teplinsky, *supra* note 30, *with* Bryan H. Choi, *NIST's Software Un-Standards*, 9 GEO. L. TECH. REV. 66, 68–69 (2025).

guidance focus on downstream vendors who provide "finished" software to end users. Much less attention is given to developers of upstream software components, namely the maintainers of open-source projects. While some lip service is paid to technical measures that could improve supply chain security, there is almost no consideration of legal accountability measures. Without the force of penalties, there will be little incentive for the open-source community to change its settled ways.

Part III unpacks conventional objections to open-source liability. The three most common arguments are: (1) charitable contributions should be immune from liability; (2) the economic value of the open-source information commons is too important to imperil; and (3) free publication of open-source code is protected by free speech values. On the surface, these claims seem to make a compelling case for legal forbearance, above and beyond the usual calculus for proprietary software. Yet, as the subsequent Section shows, the first two claims are directly refuted by available case law. The third argument is toothless in most conventional cases, although it does suggest a narrow immunity for purely academic publications of open-source code.

Because open-source contributions are a species of charitable donation, Part IV delves into the law of tainted donations. Older case law shows that tainted donations are not a novel problem in law. A close examination of two areas — food donations and blood donations — shows that policymakers have settled on negligence or gross negligence as an acceptable compromise that balances both the urgent public need for donations with the fairness to victims injured by tainted donations. More broadly, even if lawmakers disagree about the degree of fault that must be shown in order to trigger liability, there is overwhelming support for retaining a fault-based test to govern harms caused by charitable activities.

Moving to a normative lens, Part V offers a three-part proposal for open-source liability. First, the case law suggests that negligence is the right test, but that the standard of care should be set at a heightened threshold when charitable interests are involved. Either professional negligence or gross negligence would be more appropriate than ordinary negligence at balancing the competing needs for software safety and social benefit. Second, a parallel set of duties should extend to code-hosting platforms (or "code banks") such as GitHub in order to facilitate greater traceability and accountability on a systemic level. Third, an exception will likely be needed for pure speech interests — such as the open-source publishing of academic research — where the expressive interests of source code outweigh its functional aspects.

Although the analysis here focuses on conventional software, it extends by principle to open data and to open weights AI models.[32]

II. CURRENT EFFORTS

Software supply chains have long been known to be a threat vector,[33] but the discovery of the SolarWinds cyberattack in December 2020 sent the federal government's response into higher gear.[34] By May 2021, the White House issued an Executive Order on cybersecurity calling for development of extensive new guidelines on enhancing software supply chain security.[35] It followed up with a National Cybersecurity Strategy in March 2023.[36]

In accordance with the cybersecurity directive, agencies like the National Institute of Standards and Technology ("NIST"), Cybersecurity and Infrastructure Security Agency ("CISA"), and National Security Agency issued preliminary guidance on software supply chain

---

32. *See* Bryan H. Choi, *Open Source AI, Open Liability AI*, 8 J.L. & INNOVATION (forthcoming 2026); *see also* NAT'L TELECOMM. INFO. ADMIN. (NTIA), DUAL-USE FOUNDATION MODELS WITH WIDELY AVAILABLE MODEL WEIGHTS (2024), https://www.ntia.gov/sites/default/files/publications/ntia-ai-open-model-report.pdf [https://perma.cc/2EZ2-SYNE]; Office of Tech., FTC, *On Open-Weights Foundation Models*, FTC TECH. BLOG (July 10, 2024), https://www.ftc.gov/policy/advocacy-research/tech-at-ftc/2024/07/open-weights-foundation-models [https://perma.cc/AV3T-ARLG] (content removed by Trump administration, *see* Rebecca Bellan, *FTC Removes Lina Khan-Era Posts About AI Risks and Open Source*, TECHCRUNCH (Oct. 20, 2025, at 9:49 PT), https://techcrunch.com/2025/10/20/ftc-removes-lina-khan-era-posts-about-ai-risks-and-open-source/ [https://perma.cc/4T8C-AE5F]); *Open Data Program*, NIST, https://www.nist.gov/spo/open-data-program [https://perma.cc/MTX2-QSKZ].

33. *See* U.S. GOV'T ACCOUNTABILITY OFF., REPORT TO CONGRESSIONAL ADDRESSEES: FEDERAL RESPONSE TO SOLARWINDS AND MICROSOFT EXCHANGE INCIDENTS 6 n.14 (2022), https://www.gao.gov/assets/gao-22-104746.pdf [https://perma.cc/SYC3-EZTL] (collecting numerous GAO reports dating back to 2012 warning of IT supply chain risks); HERR ET AL., *supra* note 6, at 6 (documenting 115 instances of publicly reported software supply chain attacks between 2010 and 2020).

34. *See* U.S. GOV'T ACCOUNTABILITY OFF., *supra* note 33, at 1 (calling SolarWinds "one of the most widespread and sophisticated hacking campaigns ever conducted against the federal government and private sector"); *see also* David E. Sanger, Clifford Krauss & Nicole Perlroth, *Cyberattack Forces a Shutdown of a Top U.S. Pipeline*, N.Y. TIMES (May 8, 2021), https://www.nytimes.com/2021/05/08/us/politics/cyberattack-colonial-pipeline.html [https://perma.cc/N77T-XUBH] (observing that the Colonial Pipeline ransomware attack, along with the SolarWinds intrusion, "has elevated cybervulnerabilities to the top of [the White House's] national security agenda").

35. *See* Exec. Order No. 14,028 § 4, 3 C.F.R. 14028 (2021) (calling for development of guidance on, inter alia, secure software development practices, use of automated tools, use of Software Bill of Materials ("SBOM"), participation in vulnerability disclosure programs, and attestations of conformity with best practices).

36. *See* THE WHITE HOUSE, *supra* note 7. *But see* Exec. Order No. 14,306, 90 Fed. Reg. 24723 (June 6, 2025) (rescinding and curtailing portions of President Biden's executive orders on cybersecurity).

security.[37] Private entities also issued reports specific to the open-source supply chain.[38]

That guidance can be understood as moving in three major parts: (1) recommended technological practices for downstream vendors of commercial software, (2) recommended technological practices for upstream maintainers of open-source projects,[39] and (3) mandatory legal duties on downstream vendors.

Table 1: Current Guidance on Securing the Software Supply Chain

	Downstream Vendors	**Upstream Maintainers**
Technical Measures	• Software Bill of Materials • Security by design	• Authentication • Automated tools
Liability Rules	• Attestation • Duty to inspect	• [*undefined*]

This ordering reveals a conspicuous gap: none of the new guidance contemplates legal measures that shift liability onto maintainers and contributors of open-source components. This Part steps through each of the three enumerated approaches along with their shortcomings. The fourth box is the subject of the remainder of the Article.

A. Technical Measures: Downstream Vendors

The bulk of early policy attention has focused on technological remediations by downstream adopters and integrators of open-source software. One main thrust centers on eliminating "software of unknown provenance" through transparency requirements. A second thrust seeks to encourage adoption of secure-by-design practices in order to harden the substantive code deployed in critical systems.

37. *See* NIST, *supra* note 15, at 4–5 (listing "compromising open-source code" as one of the three most common techniques to execute software supply chain attacks); ENDURING SEC. FRAMEWORK, *supra* note 16, at 19 ("Developers commonly use open source code in application development, with projects potentially having multiple dependencies on open source libraries which may contain vulnerabilities.").

38. *See, e.g.*, SONATYPE 2023 REPORT, *supra* note 1; SYNOPSYS, *supra* note 1.

39. *See* EGHBAL, *supra* note 19, at 87, 90 (distinguishing "maintainers" as stewards who make decisions about the project from "contributors" who make contributions but "aren't responsible for its overall success," and explaining that maintainers are "the bottleneck to everyone else's contributions")

The most widely recognized initiative prompts software vendors to provide a Software Bill of Materials ("SBOM").[40] By providing a detailed record of the code components used to build a software application, the SBOM functions like a product's bill of materials, a shipper's bill of lading, or a food package's list of ingredients.[41] Provenance starts with tracking the use of third-party code, as well as the many dependencies contained within such code.[42] Being able to reuse free open-source code packages offers valuable efficiencies and cost savings, but it also means executing code that is of unknown origin and quality.

SBOMs are useful as a transparency process because they can provide greater visibility for identifying vulnerabilities, facilitating efficient maintenance and remediation, and ensuring components are kept up to date.[43] Moreover, publication of SBOMs serves a public accountability role by allowing third parties to audit the diligence of the software developer's maintenance efforts. If purchasers are able to differentiate higher quality code from cheap, junk-food code, then developers would have incentives to use better "ingredients."[44]

Yet, SBOMs have their limitations.[45] Like any transparency measure, SBOMs do not directly improve software safety; disclosure is not a substitute for doing the hard work of producing and maintaining good

---

40. Exec. Order No. 14,028, *supra* note 35, at §§ 4(e)(vii), 4(f) (directing NIST to develop guidance on "providing a purchaser a Software Bill of Materials (SBOM) for each product directly or by publishing it on a public website" and the National Telecommunications and Information Administration (NTIA) to "publish minimum elements for an SBOM"); *see also* Sharma, *supra* note 4, at 1198 ("The EO's most direct mandate was a requirement that software vendors provide [federal] agency customers with a Software Bill of Materials."); Trevor Stalnaker, Nathan Wintersgill, Oscar Chaparro, Massimiliano Di Penta, Daniel M German & Denys Poshyvanyk, *BOMs Away! Inside the Minds of Stakeholders: A Comprehensive Study of Bills of Materials for Software Systems*, 46 INT'L CONF. ON SOFTWARE ENG'G *1, *1 (2024).

41. *See Software Security in Supply Chains: Software Bill of Materials (SBOM)*, NIST (May 3, 2022), https://www.nist.gov/itl/executive-order-14028-improving-nations-cybersecurity/software-security-supply-chains-software-1 [https://perma.cc/2CCM-FNPC]; *see also* NTIA, U.S. DEP'T OF COMMERCE, THE MINIMUM ELEMENTS FOR A SOFTWARE BILL OF MATERIALS (SBOM) 9 (2021), https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf [https://perma.cc/7A89-54WA] (recommending that SBOMs should include standardized data fields such as the software component's name, version, dependencies, and author).

42. *See* SONATYPE 2023 REPORT, *supra* note 1, at 25 (explaining that the average Java application contains 148 dependencies, resulting in "an average of 1,500 dependency changes per year per application"); Sharma, *supra* note 4, at 1146 ("Because open source is modular, it is like a Russian doll: one project can be dependent on another project, which is dependent on a third project. Software dependencies are nested and harder to find.").

43. *See* NTIA, *supra* note 41, at 6.

44. *See* Sharma, *supra* note 4, at 1199 ("SBOMs empower agency customers to put upward pressure on software vendors to improve their security practices.").

45. *See* NTIA, *supra* note 41, at 7.

code.[46] A mere list of software components and dependencies does not indicate whether those elements are safe or secure for use, let alone whether the disclosure is truthful, accurate, or complete.[47] Compounding the problem is that the SBOM protocol remains immature, with inadequate offerings and a lack of standardization.[48] Adoption rates remain very low: Although vendors near the end of the supply chain have greater incentive to produce SBOMs, others further up the supply chain have felt little pressure to embrace the change.[49]

A different, more substantive initiative focuses on security-by-design.[50] The basic tenet is that security should be "proactive and preventive, not reactive and remedial."[51] This approach encourages software developers to revise their software development practices by embedding security principles throughout the lifecycle of design, implementation, testing, and maintenance.[52] For example, NIST has partnered with industry leaders to publish a Secure Software Development Framework ("SSDF").[53] NIST's guidance is a high-level document that

---

46. *See* Chinmayi Sharma, *Open-Source Security: How Digital Infrastructure Is Built on a House of Cards*, LAWFARE (July 25, 2022, at 08:01 ET), https://www.lawfaremedia.org/article/open-source-security-how-digital-infrastructure-built-house-cards [https://perma.cc/UY4N-9NZ5] ("But, SBOMs are insufficient . . . . An SBOM is simply a list of the ingredients, or codebases, that comprise software you purchased. It does not provide a list of vulnerabilities nor does it impose any minimum security requirements on the vendor generating the SBOM.").

47. *See* Stalnaker et al., *supra* note 40, at *8 (discussing SBOM accuracy and completeness); Santiago Torres-Arias et al., *A Viewpoint on Knowing Software Bill of Materials Quality When You See It*, IEEE SEC. & PRIVACY, Nov.–Dec. 2023, at 50, 51 ("To be blunt, there is as yet no penalty for lying.").

48. *See* Boming Xia, Tingting Bi, Zhenchang Xing, Qinghua Lu & Liming Zhu, *An Empirical Study on Software Bill of Materials: Where We Stand and the Road Ahead*, 45 INT'L CONF. ON SOFTWARE ENG'G 2630, 2630 (2023); Sabato Nocero, Simone Romano, Massimiliano Di Penta, Rita Francese & Giuseppe Scanniello, *Software Bill of Materials Adoption: A Mining Study from GitHub*, 2023 IEEE INT'L CONF. ON SOFTWARE MAINT. & EVOLUTION 39, 47.

49. *See* Stalnaker et al., *supra* note 40, at *6 ("Participants expressed that pressure to maintain SBOMs primarily targets industry and projects at the end of a supply chain, while projects near the beginning have little incentive to produce them."); Sharma, *supra* note 4, at 1199.

50. *See* Exec. Order No. 14,028, *supra* note 35, at § 4(e)(ix) (recommending "conformity with secure software development practices"); THE WHITE HOUSE, *supra* note 7, at 5 ("We must . . . embrace security and resilience by design . . . ."); NIST, *supra* note 15, at 11–13; ENDURING SEC. FRAMEWORK, *supra* note 16, at 20; CISA, SHIFTING THE BALANCE OF CYBERSECURITY RISK: PRINCIPLES AND APPROACHES FOR SECURE BY DESIGN SOFTWARE (2023), https://www.cisa.gov/sites/default/files/2023-10/SecureByDesign_1025_508c.pdf [https://perma.cc/R4SM-RFK9]; *see also* Lee A. Bygrave, *Security by Design: Aspirations and Realities in a Regulatory Context*, 8 OSLO L. REV. 126, 152–54 (2021) (situating the security-by-design movement within "a burgeoning set of 'by design' discourses" such as privacy-by-design and data-protection-by-design).

51. Renana Arizon-Peretz, Irit Hadar & Gil Luria, *The Importance of Security Is in the Eye of the Beholder: Cultural, Organizational, and Personal Factors Affecting the Implementation of Security by Design*, 48 IEEE TRANSACTIONS ON SOFTWARE ENG'G 4433, 4433 (2022).

52. *See, e.g.*, DAVID A. WHEELER, SECURE PROGRAMMING 7–8 (2015); MICHAEL HOWARD & STEVE LIPNER, SECURITY DEVELOPMENT LIFECYCLE (2009).

53. NIST, SECURE SOFTWARE DEVELOPMENT FRAMEWORK (SSDF) VERSION 1.1 (2022).

aggregates general recommendations, such as making sure to use care when selecting third-party software components[54] and verifying that those components are properly maintained.[55] Each software developer adopting the SSDF is expected to flesh out how those general recommendations are implemented to fit individual contexts. Meanwhile, entities like CISA, the Federal Trade Commission ("FTC"), and the Center for Internet Security ("CIS") have sought to define priority action items that establish a minimum baseline.[56]

But security-by-design is challenging to define.[57] There are few metrics or empirical evidence to show which security controls might be effective.[58] The standards that currently exist give software developers substantial leeway to choose for themselves how much or how little action to take based on a self-assessment of risk.[59] Meanwhile, most software developers remain unenthusiastic about integrating extra design principles into their work.[60] As a result, security-by-design often devolves into a paperwork drill that fails to produce meaningful improvements in code quality.[61] Again, vendors closer to the point of sale have greater market incentive to claim compliance with security-by-design principles than developers who are more removed.

---

54. *Id.* at 12 ("Acquire and maintain well-secured software components . . . from commercial, open-source, and other third-party developers for use by the organization's software.").

55. *Id.* at 13 ("Verify that acquired commercial, open-source, and all other third-party software components comply with the requirements, as defined by the organization, throughout their life cycles.").

56. *See Secure by Design Alerts*, CISA, https://www.cisa.gov/securebydesign/alerts [https://perma.cc/K8YT-EXY8]; Office of Tech., FTC, *Security Principles: Addressing Vulnerabilities Systematically*, FTC TECH. BLOG (Apr. 17, 2024), https://www.ftc.gov/policy/advocacy-research/tech-at-ftc/2024/04/security-principles-addressing-vulnerabilities-systematically [https://perma.cc/GJV6-CE3T]; *The 18 CIS Critical Security Controls*, CTR. FOR INTERNET SEC., https://www.cisecurity.org/controls/cis-controls-list [https://perma.cc/4J5A-7N79]; *see also* Scott J. Shackelford, Anne Boustead & Christos Makridis, *Defining "Reasonable" Cybersecurity: Lessons from the States*, 25 YALE J.L. & TECH. 86 (2023).

57. *See* Scott J. Shackelford, Craig Jackson, Scott Russell, Emily K. Adams, Anne Boustead & Christos Makridis, *The Difficulties of Defining "Secure-by-Design"*, LAWFARE (Feb. 6, 2024, at 08:00 ET), https://www.lawfaremedia.org/article/the-difficulties-of-defining-secure-by-design [https://perma.cc/CAT6-VAR3] (explaining that although security-by-design dates back to the 1970s, "the core goal of making cybersecurity central to decision-making that lies at the heart of security-by-design remains elusive, as seen in findings from a 2023 state-level survey"); Bygrave, *supra* note 50, at 155, 160 (explaining that the meaning of both "security" and "by design" are "contested and often contentious" and "afflicted by slippery semantics").

58. Shackelford et al., *supra* note 57 (lamenting that "cybersecurity standards and guidance-making bodies continue to produce long lists of must-do controls with little to no explanation of how those controls were selected, much less any evidence to support those as effective").

59. *See* Choi, *supra* note 31.

60. *See* Bygrave, *supra* note 50, at 174 (citing a recent survey of software engineers showing that "a large percentage of the respondents (40 percent) felt that it was not their responsibility to integrate privacy and security into their work and did not find pleasure in doing so").

61. *See* Choi, *supra* note 31.

B. Technical Measures: Upstream Maintainers

Much less is expected of open-source maintainers who manage code further up in the supply chain.[62] Although SBOMs and security-by-design are recommended as general good practices for all developers, policymakers express low expectations that open-source developers will voluntarily adhere to any rigorous safeguards.[63] Instead, the technical measures that have had greatest impact on open-source practices are those imposed top-down by intermediary platforms such as GitHub.[64] In particular, two interventions stand out: identity authentication and automated tools for supply chain security.

Anonymity has long been an accepted feature of open-source communities, in part because of deep roots in the hacker ethos, and in part because of core commitments to maximizing open access and participation. To be sure, reputation matters, and many contributors disclose their identity in order to reap the reputational benefits of their contributions. Individual projects can and do establish their own norms and practices. For example, some projects require in-person verification before allowing contributions, while other projects take a more open-door approach.[65] But user accounts have long been pseudonymous by default.

Beginning in 2022, most major open-source platforms announced they would require all users to verify identity using multi-factor

---

62. *See, e.g.*, NIST, SOFTWARE SUPPLY CHAIN SECURITY GUIDANCE UNDER EXECUTIVE ORDER (EO) 14028 SECTION 4E, at 2 (2022), https://www.nist.gov/system/files/documents/2022/02/04/software-supply-chain-security-guidance-under-EO-14028-section-4e.pdf [https://perma.cc/W7BP-9MZ2] (declaring that open-source software is "out of scope" of NIST's supply chain security guidance); EGHBAL, *supra* note 19, at 136 ("Bigger projects might use a monitoring tool like Snyk or SourceClear to scan their code and notify maintainers of known security vulnerabilities, but the maintainers of smaller open source projects, frankly, often can't be bothered.").

63. *See, e.g.*, CISA, CISA OPEN SOURCE SOFTWARE SECURITY ROADMAP 8 (2023), https://www.cisa.gov/sites/default/files/2023-09/CISA-Open-Source-Software-Security-Roadmap-508c%20%281%29.pdf [https://perma.cc/XPV2-9SE5] (acknowledging that there are "unique challenges to achieving comprehensive SBOM generation throughout open source supply chains"); NIST, *supra* note 53, at 1 ("Future work may expand on this publication and potentially cover topics such as . . . how the SSDF could be applied in the context of open-source software.").

64. *See* EGHBAL, *supra* note 19, at 21 ("GitHub had a meteoric impact on open source. . . . Although there's no requirement that developers *must* use GitHub to write open source software, GitHub is by far the dominant code-hosting platform today."); *id.* at 77 ("GitHub was the highway system that transformed how open source software was produced. . . . At some level, every project on GitHub now looks the same, regardless of its language or function.").

65. EGHBAL, *supra* note 19, at 46 ("Debian, an operating system based on Linux, requires that developers follow an extensive onboarding process in which they are expected to read a manual, find a mentor, and meet a maintainer in person who can vouch for their identity. On the other hand, it's common among JavaScript developers to give away commit access more freely . . . .").

authentication.[66] Observing that most software supply-chain attacks occur through compromised user accounts, GitHub and other platforms reasoned that they needed to do more to elevate minimum security measures. Several prominent members of the open-source community have opposed this change and threatened to abandon their projects as a consequence.[67] Nevertheless, the large platforms refused to back down and have begun rolling out multifactor authentication to their users.[68] Smaller platforms may face more challenges in implementing such changes.[69]

A second example is the rollout of automated tools by major open-source platforms.[70] These tools perform tasks such as vulnerability detection, code-signing of binaries, and auto-generation of dependency lists.[71] Some tools run by default regardless whether individual

66. *See* Mike Hanley, *Software Security Starts with the Developer: Securing Developer Accounts with 2FA*, GITHUB BLOG (May 6, 2022), https://github.blog/news-insights/company-news/software-security-starts-with-the-developer-securing-developer-accounts-with-2fa/ [https://perma.cc/7CJ5-9S2T]; Betty Li, *Making Popular Ruby Packages More Secure*, RUBYGEMS BLOG (June 13, 2022), https://blog.rubygems.org/2022/06/13/making-packages-more-secure.html [https://perma.cc/LB8E-WD5X]; Myles Borins, *Top-100 npm Package Maintainers Now Require 2FA, and Additional Security-Focused Improvements to npm*, GITHUB BLOG (Feb. 11, 2022), https://github.blog/security/supply-chain-security/top-100-npm-package-maintainers-require-2fa-additional-security/ [https://perma.cc/GB48-NM9Z]; Donald Stufft, *Securing PyPI accounts via Two-Factor Authentication*, PYTHON PACKAGE INDEX BLOG (May 25, 2023), https://blog.pypi.org/posts/2023-05-25-securing-pypi-with-2fa/ [https://perma.cc/Z55S-63PH].

67. *See* Sharma, *supra* note 4, at 1195 (observing that "the new requirement resulted in an outcry from community members particularly averse to top-down mandates — authors of extremely popular projects threatened to abandon their posts").

68. *See* Mike Hanley, *Securing Millions of Developers Through 2FA*, GITHUB BLOG (May 14, 2024), https://github.blog/security/supply-chain-security/securing-millions-of-developers-through-2fa/ [https://perma.cc/3KFG-YK9Y] (reporting "an opt-in rate of nearly 95% across code contributors who received the 2FA requirement in 2023" and a "54% increase in 2FA adoption among all active contributors on GitHub.com"); Mike Fiedler, *2FA Required for PyPI*, PYTHON PACKAGE INDEX BLOG (Jan. 1, 2024), https://blog.pypi.org/posts/2024-01-01-2fa-enforced/ [https://perma.cc/7XM5-FCPF].

69. *See* Sharma, *supra* note 4, at 1196 ("While GitHub might have the resources to mandate 2FA for all its users, other less-resourced entities do not.").

70. *Cf.* Exec. Order No. 14,028, *supra* note 35, at § 4(e)(iii)–(iv) (seeking NIST's guidance on "employing automated tools, or comparable processes, to maintain trusted source code supply chains, thereby ensuring the integrity of the code" and to "check for known and potential vulnerabilities and remediate them").

71. *See* Press Release, CISA Announces New Efforts to Help Secure Open Source Ecosystem (Mar. 7, 2024), https://www.cisa.gov/news-events/news/cisa-announces-new-efforts-help-secure-open-source-ecosystem [https://perma.cc/A89X-XXF6].

developers opt in.[72] Other tools are elective and depend on whether the developer chooses to activate the tool.[73]

Automated tools can offer broad benefits, but they carry at least two limitations. First, because most tools remain in early-stage development, their efficacy remains speculative. Second, most tools publicized thus far perform only narrowly scoped tasks — often in the form of providing notice or alerts — that then depend on responsive action by open-source maintainers. In other words, automation typically serves a decision-support role that does not substitute for the human decisions being made by individual maintainers and contributors.

The familiar takeaway is that platforms exert considerable power to set technology policy, particularly in decentralized settings like open-source communities.[74] The White House and federal agencies have begun nudging these platforms to take stronger prophylactic action, although the platforms' responses remain tepid and minimalist. Multifactor authentication is still one step removed from verifying a government-issued ID, let alone real-world identity. Automated tool development and adoption continue to languish. Nevertheless, the growing maturity of technical interventions triggers consequential questions whether internet intermediaries should be enlisted to control technology choices on behalf of distributed users.[75]

C. Liability Rules: Downstream Vendors

Policymakers have begun to acknowledge that new legal accountability measures are needed in addition to technological solutions. But those legal reform proposals have focused almost exclusively on downstream vendors, leaving vast gaps in oversight elsewhere along the software supply chain. Because the judiciary has been slow to heed calls

---

72. *See* Brian Fox, *New Maven Central Security Capabilities*, MAVEN CENT. REPOSITORY DOCUMENTATION (May 10, 2021), https://central.sonatype.org/news/20210510_new-security-capabilities/ [https://perma.cc/35DP-Q4NC] (announcing that all staged repositories will be automatically scanned for vulnerabilities starting May 12, 2021); *see also* RUST FOUND., SECURITY INITIATIVE REPORT 13–14 (2024), https://rustfoundation.org/wp-content/uploads/2024/06/security-initiative-report-february-2024.pdf [https://perma.cc/M4GH-6CPF] (describing several tools under development to automatically detect malicious activity).

73. *See* Luke Hinds & Hayden Blauzvern, *Sigstore: Simplifying Code Signing for Open Source Ecosystems*, OPENSSF BLOG (Nov. 21, 2023), https://openssf.org/blog/2023/11/21/sigstore-simplifying-code-signing-for-open-source-ecosystems/ [https://perma.cc/H9JD-7F8T]; Brian DeHamer & Philip Harrison, *Introducing npm Package Provenance*, GITHUB BLOG (May 12, 2023), https://github.blog/security/supply-chain-security/introducing-npm-package-provenance/ [https://perma.cc/9Z5G-SL4E].

74. *Cf.* Jonathan Zittrain, *A History of Online Gatekeeping*, 19 HARV. J.L. & TECH. 253 (2006); Jack M. Balkin, *Free Speech Versus the First Amendment*, 70 UCLA L. REV. 1206, 1216 (2023).

75. *See, e.g.*, James Grimmelmann & Pengfei Zhang, *An Economic Model of Online Intermediary Liability*, 38 BERKELEY TECH. L.J. 1011 (2023); Felix T. Wu, *Collateral Censorship and the Limits of Intermediary Immunity*, 87 NOTRE DAME L. REV. 293 (2011).

for common law tort remedies,[76] recent attention has turned instead to executive or legislative interventions. For example, commentators have hailed federal agencies like the FTC and CISA for establishing and enforcing better security practices among software companies.[77] To induce broader compliance prior to enforcement, the White House also introduced a software security attestation requirement for all federal software contractors.[78] Separately, Congress has shown bipartisan support for new legislation to shift liability onto the "final goods assemblers" of software.[79]

The FTC has positioned itself as the primary enforcer of "reasonable" or "minimum" cybersecurity measures.[80] A growing contingent of scholars has urged the FTC to act more aggressively to establish a "common law" of enforcement actions against entities with poor software practices.[81] Some commentators have praised those efforts for offering clearer norms and precedents around acceptable practices.[82]

---

76. *See* Choi, *supra* note 25; Lubin, *supra* note 28, at 1904–05 (2025); *supra* notes 23–31 and accompanying text. *But see* Tschider, *supra* note 24.

77. *See* Press Release, FTC, FTC Releases 2023 Privacy and Data Security Update (Mar. 28, 2024), https://www.ftc.gov/news-events/news/press-releases/2024/03/ftc-releases-2023-privacy-data-security-update [https://perma.cc/NF4H-BZZS] ("The FTC also has remained active in targeting companies that fail to implement reasonable data security measures to protect consumer data."); *see also* Statement, Erik Gerding, Director, SEC Div. of Corp. Fin., Cybersecurity Disclosure (Dec. 14, 2023), https://www.sec.gov/newsroom/speeches-statements/gerding-cybersecurity-disclosure-20231214 [https://perma.cc/KR4S-PZTX] (explaining new final rules requiring "public companies to disclose both material cybersecurity incidents they experience and, on an annual basis, material information regarding their cybersecurity risk management, strategy, and governance").

78. *See* Memorandum M-22-18, Shalanda D. Young, Director, Office of Mgmt. & Budget, Enhancing the Security of the Software Supply Chain through Secure Software Development Practices (Sept. 14, 2022), https://bidenwhitehouse.archives.gov/wp-content/uploads/2022/09/M-22-18.pdf [https://perma.cc/L92D-YNPU]; Jim Dempsey, Steven B. Lipner & James Andrew Lewis, *Making Attestation Work for Software Security*, LAWFARE (July 18, 2024, at 12:00 ET), https://www.lawfaremedia.org/article/making-attestation-work-for-software-security [https://perma.cc/WLY7-9M8P].

79. U.S. CYBERSPACE SOLARIUM COMM'N, *supra* note 7, at 76.

80. *See* Shackelford et al., *supra* note 56, at 99–101 (explaining the FTC's actions to regulate "reasonable" cybersecurity under its Section 5 unfairness authority); William McGeveran, *The Duty of Data Security*, 103 MINN. L. REV. 1135, 1194–96 (2019); *see also* Office of Tech., FTC, *supra* note 32 (asserting the FTC's ability and interest in regulating open-source software and open-weights AI models).

81. *See* Andrew D. Selbst & Solon Barocas, *Unfair Artificial Intelligence: How FTC Intervention Can Overcome the Limitations of Discrimination Law*, 171 U. PA. L. REV. 1023, 1044–47 (2023); Dennis D. Hirsch, *From Individual Control to Social Protection: New Paradigms for Privacy Law in the Age of Predictive Analytics*, 79 MD. L. REV. 439, 491–92 (2020); Daniel J. Solove & Woodrow Hartzog, *The FTC and the New Common Law of Privacy*, 114 COLUM. L. REV. 583, 600 (2014).

82. *See* ISABELLA WRIGHT & MAIA HAMIN, ATLANTIC COUNCIL, "REASONABLE CYBERSECURITY" IN FORTY-SEVEN CASES: THE FEDERAL TRADE COMMISSION'S ENFORCEMENT ACTIONS AGAINST UNFAIR AND DECEPTIVE CYBER PRACTICES 2 (2024), https://dfrlab.org/wp-content/uploads/sites/3/2024/06/47-cases-ftc-cyber-csi.pdf [https://perma.cc/C5MT-QJWT] (arguing that the FTC "has effectively constructed a body of 'reasonable' cybersecurity practices and clear precedent for their enforcement").

Yet, because the FTC's regulatory mission is focused on business entities that sell consumer-facing goods or services,[83] its oversight overlooks most open-source offerings, many of which are not organized as business entities or engaged in any commercial sales. Moreover, the FTC's resources are limited,[84] and recent judicial decisions and executive retrenchments have further called into question the ambit of the FTC's common law paradigm.[85]

Attestation is the other leading executive initiative, which requires an affirmative statement by software vendors that they have adhered to minimum secure software development requirements.[86] This requirement provides regulators with greater ability to demand compliance with the technical measures described above. For federal contractors, these attestations have been mandatory for any software developed after September 14, 2022.[87] For other entities, CISA has launched a voluntary "pledge" initiative that performs a similar function, albeit without any binding legal effect.[88] In theory, penalties for false attestation could include bid protests, contract terminations, and debarment

---

83. *See Mission*, FTC, https://www.ftc.gov/about-ftc/mission [https://perma.cc/C8KT-PWWN] ("The FTC's mission is protecting the public from deceptive or unfair *business practices* and from unfair *methods of competition* through law enforcement, advocacy, research, and education." (emphases added)).

84. *See* WRIGHT & HAMIN, *supra* note 82, at 28 (observing that with only forty-seven total cases, "the FTC has typically resolved only a few cyber enforcement cases each year").

85. *See* Shackelford et al., *supra* note 56, at 100–01 (citing LabMD Inc. v. FTC, 894 F.3d 1221 (11th Cir. 2018)); SEC v. Jarkesy, 603 U.S. 109, 127 (2024) ("The Constitution prohibits Congress from 'withdraw[ing] from judicial cognizance any matter which, from its nature, is the subject of a suit at the common law.'" (alteration in original) (citation omitted)); Loper Bright Enters. v. Raimondo, 603 U.S. 369 (2024) (overturning *Chevron* deference); *see also* THE WHITE HOUSE, AMERICA'S AI ACTION PLAN 3–4 (2025), https://www.whitehouse.gov/wp-content/uploads/2025/07/Americas-AI-Action-Plan.pdf [https://perma.cc/U7TE-46Z8] (ordering a review of all FTC investigations, final orders, consent decrees, and injunctions "to ensure that they do not advance theories of liability that unduly burden AI innovation"); Justin (Gus) Hurwitz, *Data Security and the FTC's UnCommon Law*, 101 IOWA L. REV. 955, 984–88 (2016). *But see* Tschider, *supra* note 24, at 647, 662 (explaining that FTC consent decrees have had an influence on state common law tort actions).

86. *See* Memorandum M-23-16, Shalanda D. Young, Director, Office of Mgmt. & Budget, Update to Memorandum M-22-18, Enhancing the Security of the Software Supply Chain through Secure Software Development Practices (2023), https://www.whitehouse.gov/wp-content/uploads/2023/06/M-23-16-Update-to-M-22-18-Enhancing-Software-Security.pdf [https://perma.cc/XG3D-Z485] ("An attestation provided by [the producer of the software end product] to an agency serves as an affirmative statement that the producer follows the secure software development minimum requirements, as articulated in the common form [released by CISA].").

87. *Id.*; *see also* Dempsey et al., *supra* note 78 (explaining that this process went into effect for "critical software" on June 11, 2024, and goes into effect for other software on September 11, 2024).

88. *See Secure by Design Pledge*, CISA, https://www.cisa.gov/securebydesign/pledge [https://perma.cc/8RKW-4CFP].

from future federal procurements, as well as hefty monetary penalties under the False Claims Act.[89]

Here, too, the focus remains on vendors of downstream deliverables, and the promise being extracted is surprisingly weak: The three-page attestation form soft-pedals the minimum practices required to be in compliance.[90] Generous exemptions are allowed for vendors who cannot comply with even those weakened expectations.[91] Efforts to require stronger assurances have been opposed and retracted.[92] Moreover, past experience suggests that software contractors face little to no consequence for issuing specious attestations without actually complying with the required controls.[93]

A more weighty proposal comes from the bipartisan U.S. Cyberspace Solarium Commission, which recommends shifting liability onto the "final goods assemblers" of software.[94] The Commission's report explains that users are entirely reliant on software vendors, and that a liability rule would encourage final goods assemblers to be more timely

89. *See* Lipner, *supra* note 29; *see also* Benjamin J. McMichael, Mackenzi Barrett & W. Kip Viscusi, *A Constitutional False Claims Act*, 102 WASH. U. L. REV. 677, 679–80 (2025) (explaining that the False Claims Act can be a significant source of liability, because it allows "private parties to pursue *qui tam* suits . . . with no requirement of government oversight," and that the severity of sanctions including "treble damages and penalties add[s] up quickly, providing obvious incentives for defendants to settle").

90. *See Secure Software Development Attestation Form Version 1.0*, CISA (2024), https://www.cisa.gov/sites/default/files/2024-04/Self_Attestation_Common_Form_FINAL_508c.pdf [https://perma.cc/2JFV-CJZL] (requiring compliance with four requirements: (1) use of secure software development environments; (2) good-faith effort to maintain trusted source code supply chains; (3) maintenance of provenance information; and (4) use of automated vulnerability scanning tools); Dempsey et al., *supra* note 78 ("The attestation form is also riddled with caveats not found in the SSDF. For example: credentials need be encrypted only 'to the extent feasible'; developers need only make a 'good-faith effort' to maintain trusted source code supply chains; and SBOMs must be maintained 'to the greatest extent feasible.'").

91. *See* Dempsey et al., *supra* note 78 (explaining that developers who cannot attest to the minimum practices must submit a "Plan of Action & Milestones," but they are not required to set a deadline or "ever come up to speed").

92. *See* Exec. Order No. 14,306, *supra* note 36 (striking heightened attestation requirements instituted by Executive Order 14,144 § 2(a)–(b)); Press Release, The White House, Fact Sheet: President Donald J. Trump Reprioritizes Cybersecurity Efforts to Protect America (June 6, 2025), https://www.whitehouse.gov/fact-sheets/2025/06/fact-sheet-president-donald-j-trump-reprioritizes-cybersecurity-efforts-to-protect-america/ [https://perma.cc/XV58-65G3] (asserting the need to eliminate "unproven and burdensome software accounting processes that prioritized compliance checklists over genuine security investments").

93. *See* Dempsey et al., *supra* note 78 (describing persistent failures in an analogous cybersecurity attestation process at the Department of Defense).

94. *See* U.S. CYBERSPACE SOLARIUM COMM'N, *supra* note 7, at 5, 76; *see also id.* at 77 (defining "final goods assembler" as "the entity that enters into an end user license agreement with the user of the product or service and is most responsible for the placement of a product or service into the stream of commerce"); *cf.* RESTATEMENT (SECOND) OF TORTS § 395 cmt. g (A.L.I. 1965) (stating that the final assembler "should have sufficient technical knowledge" and must exercise "something more than a mere inspection" when "selecting raw material and parts to be incorporated in the finished article"); *id.* § 402A cmt. q (addressing but expressing no opinion whether strict products liability should shift to the final assembler).

in producing security patches.[95] Although the Commission anticipated that this proposal would be one of the most challenging initiatives in its report,[96] the recommendation has gained traction in the upper echelons of government.[97] Recent efforts have explored ways to draft legislation that defines an appropriate standard of care.[98]

Yet, the Commission's focus on "final goods assemblers" conspicuously excludes consideration of liability for upstream software components.[99] At common law, component sellers are not automatically exempt from liability for flaws in the finished product.[100] To be sure, shielding the component seller can be sensible in simple cases.[101] But courts and commentators have long recognized the need for a more nuanced rule.[102] The Third Restatement of Products Liability suggests that distributors of component parts should remain liable where (1) the component is defective, or (2) the seller or distributor of the component "substantially participates in the integration of the component" into the final product.[103] More broadly, some commentators favor a "cheapest

95. *See* U.S. CYBERSPACE SOLARIUM COMM'N, *supra* note 7, at 76.

96. *See* U.S. CYBERSPACE SOLARIUM COMM'N, 2021 ANNUAL REPORT ON IMPLEMENTATION 28 (2021), https://cybersolarium.org/wp-content/uploads/2022/05/2021-Annual-Report-on-Implementation.pdf [https://perma.cc/ZPZ3-PULF] ("The Commission expected and has encountered significant pressure against this recommendation, which is one of the four that face known significant barriers to implementation.").

97. *See* THE WHITE HOUSE, *supra* note 7, at 21 ("The Administration will work with Congress and the private sector to develop legislation establishing liability for software products and services.").

98. *See supra* notes 23–31 and accompanying text.

99. *See* U.S. CYBERSPACE SOLARIUM COMM'N, *supra* note 7, at 76–77.

100. *See* Edward M. Mansfield, *Reflections on Current Limits on Component and Raw Material Supplier Liability and the Proposed Third Restatement*, 84 KY. L.J. 221, 230–31 (1995) (collecting case law and stating that component suppliers are not held liable under four conditions: (1) the component "is not designed for use in a particular type of finished product"; (2) it is "a standard item that is generally safe as a 'building block'"; (3) it was "adapted by another entity to manufacture a finished product"; and (4) any danger is due to "specialized end-use" rather than "normal handling and use").

101. *See* RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 5 cmt. a (A.L.I. 1998) ("As a general rule, component sellers should not be liable when the component itself is not defective as defined in this Chapter. If the component is not itself defective, it would be unjust and inefficient to impose liability . . . ."); David A. Fischer, *Product Liability: A Commentary on the Liability of Suppliers of Component Parts and Raw Materials*, 53 S.C. L. REV. 1137, 1140–41 (2002); Mansfield, *supra* note 100, at 244 ("Where the component or raw material has multiple end-uses and no inherent danger, it may be very expensive for its supplier to gather and disseminate accurate information about potential end-use-specific risks.").

102. *See* Mansfield, *supra* note 100, at 230–31 (collecting case law and stating that component suppliers are not held liable when four conditions are met: (1) the component "is not designed for use in a particular type of finished product"; (2) it is "a standard item that is generally safe as a 'building block'"; (3) it was "adapted by another entity to manufacture a finished product"; and (4) any danger is due to "specialized end-use" rather than "normal handling and use").

103. RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 5 (A.L.I. 1998); *see also* Mansfield, *supra* note 100, at 249, 251–53 (arguing successfully that the Third Restatement's component parts rule should incorporate Restatement (Second) of Torts § 402A cmt. p, "the so-called 'substantial change' comment").

cost avoider" test that holds component manufacturers responsible in situations where they have greater expertise than the assembler, and it is more practical for the component manufacturer to detect or correct the error.[104] In fact, because of the way software is assembled and maintained, it is often the case that upstream maintainers of open-source software components will have the most intimate knowledge of how their code operates as well as how it breaks. Regardless whether software is a "product" for purposes of products liability,[105] the underlying rationale for dividing responsibility along the supply chain remains theoretically sound.

In sum, efforts to construct liability rules for downstream software vendors continue to present substantial unresolved challenges, while also placing an awkward spotlight on whether open-source components should be automatically exempt from the liability calculus.

III. WHY NOT OPEN-SOURCE LIABILITY?

The intuition that open-source software merits special treatment can be parsed into at least three modes of argument.

First is the argument from charity, namely that open-source contributions are voluntary donations of time and effort[106] and therefore should be accepted "as is" without ordinary expectations of quality. Second is the argument from innovation: Open source is a unique form of knowledge commons that generates tremendous societal value and would be devastated by excess legal costs. Third is the argument from liberty: open-source distribution is an act of information sharing guarded by fundamental rights of free expression. Moreover, the open-source community itself represents an ideological movement committed to access to knowledge and freedom to tinker.

This Part elaborates each of these objections in turn. The next Part responds by drawing on the law of tainted donations to show why each objection fails to preempt safety and liability concerns.

---

104. *See* Fischer, *supra* note 101, at 1145–50 (arguing in favor of the balancing test articulated in Verge v. Ford Motor Co., 581 F.2d 384 (3d Cir. 1978), which weighs (1) trade custom, (2) relative expertise, and (3) practicality); Richard D. Cunningham, Comment, *Apportionment Between Partmakers and Assemblers in Strict Liability*, 49 U. CHI. L. REV. 544, 549 (1982) (advocating for a "cheapest cost avoider" test). *But see* Mansfield, *supra* note 100, at 245 (arguing that a brightline test is preferable to a balancing test).

105. *See generally* Lubin, *supra* note 28 (explaining that the treatment of software as a "product" remains an unsettled issue).

106. For extended discussions of "what is open source," see Sharma, *supra* note 4, at 1138–65; CHARLES M. SCHWEIK & ROBERT C. ENGLISH, INTERNET SUCCESS: A STUDY OF OPEN-SOURCE SOFTWARE COMMONS (2012).

A. Free Participation

Perhaps the most basic and powerful intuition against open-source liability is that the open-source community is composed of volunteer hobbyists who are donating their time and work product to serve the public interest.[107] The founding principle of open-source software is that everyone should be allowed to participate in the use, study, and modification of software.[108] The simple idea that open access leads to better software has had profound influence and staying power.[109]

The nonprofit character of many open-source projects makes them appear distinguishable from commercial software entities. Many open-source projects are created for personal or experimental use, not necessarily for commercial use. The average contributor is unlikely to be incorporated, or to have liability insurance or adequate assets to compensate victims. Often, code is published and released for idle purposes as part of a communitarian ethos of information sharing. Other times, open-source projects help promote civic-minded virtues such as interoperability and open standards.[110] Many commentators have celebrated the generative serendipity that emerges from these "unfiltered contributions from broad and varied audiences" as the most cherished characteristic of software and the internet.[111] It seems churlish to sue those volunteers when they could have easily refused to make any contributions in the first place. In other areas of law such as intellectual property[112] or data privacy,[113] courts have granted more breathing room to non-commercial uses. Likewise, in tort law, legislatures have

---

107. *See* Yochai Benkler, *Coase's Penguin, or, Linux and* The Nature of the Firm, 112 YALE L.J. 369, 424–25 (2002) (theorizing that contributors to commons-based peer production are motivated by "the pleasure of creation," reputation gains, and upskilling rather than monetary incentives); SCHWEIK & ENGLISH, *supra* note 106, at 47 (observing that "the majority of OSS [open-source software] developers are not paid").

108. *See* STEVEN WEBER, THE SUCCESS OF OPEN SOURCE 62, 144 (2004) ("The key element of the open source process, as an ideal type, is voluntary participation and voluntary selection of tasks."); *see also* RICHARD STALLMAN, FREE SOFTWARE, FREE SOCIETY 20 (Joshua Gay ed., GNU Press 2002) (discussing the freedoms to run, modify, and distribute software).

109. *See* ERIC S. RAYMOND, THE CATHEDRAL AND THE BAZAAR (Tim O'Reilly ed., 2001) (1999); Sharma, *supra* note 4, at 1142–43.

110. *See* SCHWEIK & ENGLISH, *supra* note 106, at 25–26.

111. JONATHAN ZITTRAIN, THE FUTURE OF THE INTERNET — AND HOW TO STOP IT 70 (2008).

112. *See generally* Jessica Litman, *Lawful Personal Use*, 85 TEXAS L. REV. 1871, 1893 (2007) ("It would plainly be unconstitutional to prohibit a person from singing a copyrighted song in the shower or jotting down a copyrighted poem he hears on the radio."); Lee Ann Lockridge, *When is a Use in Commerce a Noncommercial Use?*, 37 FLA. ST. U. L. REV. 337 (2010). *But see* Katherine J. Strandburg, *What Does the Public Get? Experimental Use and the Patent Bargain*, 2004 WIS. L. REV. 81, 93 (noting that the experimental use exception in patent law is de minimis).

113. *See* California Consumer Privacy Act, Cal. Civ. Code § 1798.140(d) (2018) (imposing obligations on only for-profit entities).

sometimes enacted limited statutory protections in recognition of the special role of volunteers and Good Samaritans.[114] Arguably, a similar halo effect should extend to charitable donations of code.[115]

Relatedly, a second intuition is based on the contractarian principle that you get what you pay for, and that beggars cannot be choosers.[116] When the price of something is free, one might assume that it comes "as is" and there is no refund if the thing given away fails to meet expectations. When early free software pioneers developed "copyleft" license agreements that govern downstream use and modification of copyrighted code, they sought to make that assumption explicit through contractual language.[117] Today, essentially all software licenses include terms providing waivers of warranty, limitations of liability, and "as is" clauses.[118] In case the fact of no payment were not adequate warning on its own, these legal disclaimers serve notice that there is no contractual promise of quality on offer.

A third intuition that emerges from the free participation principle is the "hacker ethos," which embeds a set of structural assumptions why normal social and legal rules should not — or cannot — apply.[119] In

---

114. *See* DAN B. DOBBS ET AL., HORNBOOK ON TORTS 600 (2d ed. 2016) (describing state Good Samaritan statutes as covering "only particular charities or acts of charity"); *see also* Volunteer Protection Act of 1997, Pub. L. No. 105-19, 111 Stat. 218 (codified at 42 U.S.C. §§ 14501–14503) (limiting liability for individual volunteers of nonprofit organizations, though not for the nonprofit organizations themselves).

115. *See* Jonathan Zittrain, *Normative Principles for Evaluating Free and Proprietary Software*, 71 U. CHI. L. REV. 265, 286 (2004).

116. *See* Chong, *supra* note 25 ("[M]uch software is free. This is a problem under contract law because courts will not hold software providers liable for harms brought about for products or services for which users did not offer some form of payment — or what lawyers call 'consideration.'"); Bruce Schneier, *Software Liabilities and Free Software*, SCHNEIER ON SEC. (July 28, 2008, at 14:42 ET), https://www.schneier.com/blog/archives/2008/07/software_liabil.html [https://perma.cc/6YT2-CHLT] ("Free software wouldn't fall under a liability regime because the writer and the user have no business relationship; they are not seller and buyer."). *But see* Jacobsen v. Katzer, 535 F.3d 1373, 1379 ("The lack of money changing hands in open source licensing should not be presumed to mean that there is no economic consideration, however. There are substantial benefits, including economic benefits, to the creation and distribution of copyrighted works under public licenses that range far beyond traditional license royalties."); *see also* Robert A. Hillman & Maureen A. O'Rourke, *Rethinking Consideration in the Electronic Age*, 61 HASTINGS L.J. 311, 313–15 (2009).

117. *See, e.g.*, Lawrence Lessig, *Re-Crafting a Public Domain*, 18 YALE J.L. & HUMANITIES 56, 77–78 (2006); Michael J. Madison, *Reconstructing the Software License*, 35 LOY. U. CHI. L.J. 275, 283 (2003); Ira V. Heffen, Note, *Copyleft: Licensing Collaborative Works in the Digital Age*, 49 STAN. L. REV. 1487, 1507 (1997). *But see* EGHBAL, *supra* note 19, at 30, 33 (observing that "[t]oday, the MIT license is, by far, the most popular license used by GitHub projects," and the "fact that modern developers slap an MIT license onto their projects unthinkingly — if they bother to license their projects at all — separates them from early open source advocates").

118. *See* U.C.C. § 2-316 (A.L.I. & UNIF. L. COMM'N 1951) (allowing sellers of goods to disclaim all implied warranties of merchantability and fitness).

119. *See* STEVEN LEVY, HACKERS: HEROES OF THE COMPUTER REVOLUTION 26–30 (1984); *see also* PEKKA HIMANEN, THE HACKER ETHIC AND THE SPIRIT OF THE INFORMATION

particular, the governance structure of open-source projects supports anonymity by default.[120] Open-source platforms like GitHub use version control systems that track and attribute each pull request or code merge to a named user,[121] and they have begun to roll out multifactor authentication to protect users against account takeovers.[122] But because the open-source ideology seeks to maximize access and participation, GitHub and other platforms do not require user accounts to be traceable to real-world identities.[123] This practice also reflects the cultural legacy of the early internet, when trust among users was high and strong anonymity norms prevailed.[124] Yet, that commitment to participation maximalism appears to be correlated with more vulnerabilities and worse coding practices, and creates a set of security and evidentiary challenges unique to open source.[125]

B. Free Production

The second impediment to open-source liability is the claim that it contributes high or unique social value. What rocketed open source from fringe movement to mainstream mainstay has been the economic argument that commons-based peer production is superior to

---

AGE xvii (2001) (stating that the highest motivation for hackers to do something "is that they find it to be very interesting"); Karim R. Lakhani & Robert G. Wolf, *Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects*, *in* PERSPECTIVES ON FREE AND OPEN SOURCE SOFTWARE 3, 14 (Joseph Feller et al. eds., 2005) (finding eighty-three percent of survey respondents agree that "the hacker community is a primary source of their identity").

120. *See* Sharma, *supra* note 4, at 1145–46; *see also* SCHWEIK & ENGLISH, *supra* note 106, at 131 (noting that there is "no census of the universal population of open-source developers").

121. *See* SCOTT CHACON & BEN STRAUB, PRO GIT 40–43 (2d ed. 2014), https://git-scm.com/book/en/v2/Git-Basics-Viewing-the-Commit-History [https://perma.cc/35N9-W2AR].

122. *See supra* notes 66–69 and accompanying text.

123. *Cf.* Artur Pericles L. Monteiro, *Anonymity, Identity, and Lies*, 4 J. FREE SPEECH L. 551, 557 (2024) (observing that real-name policies can have disproportionate deterrent effects on online participation from marginalized communities); Margot Kaminski, *Real Masks and Real Name Policies: Applying Anti-Mask Case Law to Anonymous Online Speech*, 23 FORDHAM INTELL. PROP. MEDIA & ENT. L.J. 815, 877 (2013) (distinguishing "real name" policies from "traceability" policies).

124. *See* Anne Wells Branscomb, *Anonymity, Autonomy, and Accountability: Challenges to the First Amendment in Cyberspaces*, 104 YALE L.J. 1639 (1995); Monteiro, *supra* note 123, at 559 (recounting that "the early days of computers and the internet were marked by identifiers other than real names"); Bryan H. Choi, *The Anonymous Internet*, 72 MD. L. REV. 501, 542 (2013).

125. *See* SYNOPSYS, *supra* note 1, at 7 (2024) (finding that all ten of the top ten open source vulnerabilities were written in Javascript); Eric Brewer, Rob Pike, Abhishek Arya, Anne Bertucio & Kim Lewandowski, *Know, Prevent, Fix: A Framework for Shifting the Discussion Around Vulnerabilities in Open Source*, GOOGLE SEC. BLOG (Feb. 3, 2021), https://security.googleblog.com/2021/02/know-prevent-fix-framework-for-shifting.html [https://perma.cc/4YBJ-HBHH] (arguing that "owners and maintainers of critical software must not be anonymous").

proprietary modes of industrial production.[126] There are at least three forms of this argument: an earlier pair of moves rooted in incentive theory and information commons theory, which emphasizes open source's unique value, and a newer move based on incumbent power and switching costs.

The original incentive-based argument, recited at length in prior literature, plays up the comparative virtues of decentralized peer production over centralized planning.[127] The core insight is that the open-source community elicits highly motivated volunteers to make many small contributions, which otherwise might be pushed to the fringes and overlooked.[128] The efficiency gain is both quantitative and qualitative. Famously, more eyeballs can lead to more scrutiny and to more follow-on contributions.[129] But the argument goes a step further: whereas mass participation tends to create collective action problems, open-source platforms succeed because they organically match individual talent with the best uses of that effort.[130]

To explain this counterintuitive success, the early literature leans heavily on an incentives-based view of peer production. Especially touted are the virtues of nonpecuniary motivations often ignored or suppressed, such as the "urge to create" and reputational gains.[131] Those

126. *See* SAMIR CHOPRA & SCOTT D. DEXTER, DECODING LIBERATION 20–21 (2008) ("[T]he free software movement was founded on a perspective of software as a social good; the open source movement exists to make the case that open source software is more technically efficient and therefore could create more value for commercial software enterprises."); David McGowan, *Legal Implications of Open-Source Software*, 2001 U. ILL. L. REV. 241, 262 (2001) (describing the transition from "free software" to "open-source software" as a rejection by "pragmatist hackers" of anticommercial attitudes); WEBER, *supra* note 108, at 114 (explaining that broader mainstream adoption of open-source software depended on appealing to commercial and corporate interests).

127. *See* Benkler, *supra* note 107, at 375, 381; McGowan, *supra* note 126, at 251 (observing that the open source model "seems a tribute to Hayekian localized knowledge and decentralization" and "relies at least in part on the cost advantage of free labor"); Lior Strahilevitz, *Wealth Without Markets?*, 116 YALE L.J. 1472, 1485 (2007) (reviewing YOCHAI BENKLER, THE WEALTH OF NETWORKS: HOW SOCIAL PRODUCTION TRANSFORMS MARKETS AND FREEDOM (2006)) ("Benkler was the first scholar to realize that . . . [s]ome resources can be produced most efficiently neither in-house nor by an outsourced firm, but by a large group of like-minded altruists, voluntarily contributing to their creation.").

128. *See* Josh Lerner & Jean Tirole, *Some Simple Economics of Open Source*, 50 J. INDUS. ECON. 197, 204 (2002); Tim O'Reilly, *Lessons from Open-Source Software Development*, COMMC'NS ACM, Apr. 1999, at 33, 35 ("It is precisely because open source gives individuals the power to attack small problems that it is able to create unexpected innovations.").

129. *See* RAYMOND, *supra* note 109, at 30 (famously stating that "[g]iven enough eyeballs, all bugs are shallow"); Benkler, *supra* note 107, at 434 ("Given a sufficiently large number of contributions, direct monetary incentives necessary to bring about contributions are trivial.").

130. *See* Benkler, *supra* note 107, at 414, 422 ("It is not only, or even primarily, that more people can participate in production. *The widely distributed model of information production will better identify who is the best person to produce a specific component of a project*.").

131. *See id.* at 424–25; *see also* Eric von Hippel & Georg von Krogh, *Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science*, 14 ORG. SCI. 209, 217 (2003).

forces are facilitated by the internet, which dramatically lowers coordination costs.[132] Many scholars have identified community engagement and growth as critical ingredients to open-source success.[133] Much of this early literature portrays peer production as a nimbler counterweight to crude profit-maximizing incentives.[134] The success of open source is presented as a morality tale about the power of human altruism, creativity, and communitarianism.[135]

In subsequent years, that utopian view of peer production has paled.[136] Commentators have pointed out that most open-source participants are not especially altruistic but instead follow conventional models of rational, self-interested behavior.[137] Other voices have criticized the "more eyeballs" maxim as a myth that mistakes good intentions for good software development processes.[138] In fact, the vast majority of

---

132. *See* Benkler, *supra* note 107, at 404.

133. *See* SCHWEIK & ENGLISH, *supra* note 106, at 307 (finding that "in OSS commons, having more participants appears to be a positive factor for improving the software and sustaining the collective action").

134. *See* LAWRENCE LESSIG, THE FUTURE OF IDEAS: THE FATE OF THE COMMONS IN A CONNECTED WORLD 88–89 (2001); CHOPRA & DEXTER, *supra* note 126, at 17 (providing a Marxist analysis of software production).

135. *See* LESSIG, *supra* note 134, at 11 ("This is a struggle about an ideal — about what rules should govern the freedom to innovate. I would call it a 'moral question,' but that sounds too personal, or private.").

136. *See, e.g.*, Amy Kapczynski, *The Law of Information Capitalism*, 129 YALE L.J. 1460, 1493 (2020) (reviewing SHOSHANA ZUBOFF, THE AGE OF SURVEILLANCE CAPITALISM: THE FIGHT FOR A HUMAN FUTURE AT THE NEW FRONTIER OF POWER (2019) and JULIE E. COHEN, BETWEEN TRUTH AND POWER: THE LEGAL CONSTRUCTIONS OF INFORMATIONAL CAPITALISM (2019)) ("The dream that open software could free us all and that one could 'hack' the broader sociopolitical system by demanding openness at a certain technological layer, now seems painfully, obviously wrong.").

137. *See, e.g.*, WEBER, *supra* note 108, at 130–33 (dismissing altruism and self-organization as "myths" about open source that "get in the way of moving toward analytically rigorous answers"); Klaus M. Schmidt & Monika Schnitzer, *Public Subsidies for Open Source? Some Economic Policy Issues of the Software Market*, 16 HARV. J.L. & TECH. 473, 485 (2003) ("There is no reason to believe that the software industry differs fundamentally from all other industries when it comes to incentives for innovation."); Steven A. Hetcher, *Hume's Penguin, or, Yochai Benkler and the Nature of Peer Production*, 11 VAND. J. ENT. & TECH. L. 963, 983–93 (2009); Sharma, *supra* note 4, at 1154–56; von Hippel & von Krogh, *supra* note 131, at 217 ("Programmers contribute freely to the provision of [an OSS commons] because they garner private benefits from doing so."). *But see* SCHWEIK & ENGLISH, *supra* note 106, at 306 (finding "somewhat surprisingly, that the idea of demonstrating programming skill to the broader community and financial benefit appear to be less significant motivations" for OSS participation).

138. *See, e.g.*, HOWARD & LIPNER, *supra* note 52, at 18–21:

> The concept of 'Given enough eyeballs, all bugs are shallow' is wrong on many fronts: it assumes that people reviewing the code are motivated to review the code, that the people doing the reviews know what security bugs are, and that there is a critical mass of informed and motivated reviewers. But more important, . . . it just misses the point altogether . . . . Until the development processes improve in the open-source community, no major decrease in the staggering number of security bugs will occur.

peer contributions is low-quality, and what really drives the success of open source are the disproportionate efforts of a few dedicated maintainers.[139] Recent trends in related commentary have decried the overproduction of information and advocated for a return to *more* friction.[140] Nevertheless, the narrative that peer production is more efficient at producing high-quality software continues to hold significant appeal and sway.

A second set of utility-based arguments shifts away from behavioral theories of individual developers to the stored value of open-source repositories as an innovation commons.[141] Reuse of shared code, packages, and libraries is a central norm of software development that allows developers to build on the work of others.[142] Thus, the pooling of open-source contributions produces a public resource that generates

---

SCHWEIK & ENGLISH, *supra* note 106, at 75, 170–71 (observing that "field research on OSS suggests that Linus's law ['more eyeballs'] does not apply often" because "a majority of OSS projects have small teams," but still finding that larger teams and larger user communities contribute to growth stage project success); Sharma, *supra* note 4, at 1184; *see also* Zittrain, *supra* note 115, at 282 (noting that the debate is difficult to resolve and "in large part turns on which empirical examples are chosen to support each side").

139. *See* WEBER, *supra* note 108, at 65 ("[T]he success of open source cannot simply depend on getting more people or even the 'right' people to contribute to the project. It depends also, and crucially, on how those people are organized." (emphasis omitted)); Sharma, *supra* note 4, at 1160 (noting that at least twenty-three percent of open source projects have only one developer, and ninety-four percent of projects have fewer than ten developers); SCHWEIK & ENGLISH, *supra* note 106, at 180 (reporting data from 2009 that "about 70 percent of the hosted projects [on SourceForge] list only one developer as a project member" and that "[a]bout 84 percent of the projects have fewer than three developers"); *id.* at 308 (finding that "the benevolent dictator model dominates in OSS," namely that "a designated leader . . . makes the bulk of the major project decisions"); *see also* EGHBAL, *supra* note 19, at 9 (observing that one of the biggest problems facing open source maintainers is that "there are *too* many contributors — or they're the wrong kind of contributors").

140. *See* EGHBAL, *supra* note 19, at 159, 176 ("Instead, I'd flip the question to ask: *What if, rather than being underproduced, software is actually overproduced?*"); Paul Ohm & Jonathan Frankle, *Desirable Inefficiency*, 70 FLA. L. REV. 777, 804 (2018) (defining "desirable inefficiency" as introducing inefficiency in one area "in order to address a different, related enhanced problem"); Brett Frischmann & Paul Ohm, *Governance Seams*, 37 HARV. J.L. & TECH. 1117, 1120 (2023) ("Friction-in-design regulation should be a component of governance systems for the digitally networked world."); William McGeveran, *The Law of Friction*, 2013 U. CHI. LEGAL F. 15, 39 (2013) (discussing the "major concerns about frictionless sharing"); *see also* Strahilevitz, *supra* note 127, at 1493–94 (anticipating the problem of "spammers, flamers, trolls, and know-nothings").

141. *See* LESSIG, *supra* note 134, at 57 ("This free code builds a commons. This commons in turn lowers the cost of innovation. New projects get to draw upon this common code; every project need not reinvent the wheel. The resource thus fuels a wide range of innovation that otherwise could not exist.").

142. *See* Zittrain, *supra* note 115, at 277 (noting that the open source model benefits innovation because "a great deal of software is built literally on predecessors' code"); EGHBAL, *supra* note 19, at 140 ("It's cheaper to reuse existing software components than to write code from scratch . . . . The entire software industry owes its financial success to leveraging this arbitrage.").

positive spillover effects in knowledge and innovation.[143] Having this commons can reduce inefficient costs of development such as duplication of effort, interoperability obstacles, personnel turnover, and proprietary licensing fees.[144]

To be sure, the value of the open-source commons is weighed down by angst about the "tragedy of the commons."[145] The classic tragedy is where a natural resource, such as a fishery or town green, is depleted faster than it can be replenished. Likewise, the open-source commons is only as good as the quality of its contents. Here, too, the early commentary contrasts with the later commentary.

Early advocates expressed optimism that the open-source commons could be self-sustaining as long as it was left alone. After all, they explained, software cannot suffer from over-consumption because information cannot be depleted the way natural resources can.[146] The risk instead is one of under-production due to free-riding.[147] Yet, the surprising novelty of open source was that it appeared to have solved the problem of replenishment by enforcing an ethos of reciprocity.[148] Thus,

---

143. *See* Michael J. Madison, Brett M. Frischmann & Katherine J. Strandburg, *Constructing Commons in the Cultural Environment*, 95 CORNELL L. REV. 657, 668–69, 672 (2010); *see also* Brett M. Frischmann & Mark A. Lemley, *Spillovers*, 107 COLUM. L. REV. 257, 258 (2007).

144. *See* NADIA EGHBAL, ROADS AND BRIDGES: THE UNSEEN LABOR BEHIND OUR DIGITAL INFRASTRUCTURE 23 (2016) ("Free software makes it exponentially cheaper and easier to build software."); WEBER, *supra* note 108, at 245 (observing that open source reduces transaction costs associated with "the tragedy of the anticommons"); James Bessen, *Open Source Software: Free Provision of Complex Public Goods*, *in* THE ECONOMICS OF OPEN SOURCE SOFTWARE DEVELOPMENT 57, 62, 76 (Jürgen Bitzer & Philipp J. H. Schröder eds., 2006); Sharma, *supra* note 4, at 1140–42.

145. *See* LESSIG, *supra* note 134, at 22 ("This 'tragedy' consumes talk about 'the commons.' 'Ruin' is taken for granted as the destiny of those who believe in the 'freedom of the commons.'").

146. *Id.* at 68 ("Open code creates a commons; but the problem with this sort of commons is not the problem of overgrazing."); WEBER, *supra* note 108, at 154 (arguing that open-source software is "not simply a nonrival good" but "is actually antirival in the sense that *the system as a whole positively benefits from free riders*"); *see also* SCHWEIK & ENGLISH, *supra* note 106, at 7 ("In OSS commons, groups act collectively to produce public goods (i.e., software codes or open content) rather than overuse the resource, as often is the case in environmental commons.").

147. *See* Brett M. Frischmann, *An Economic Theory of Infrastructure and Commons Management*, 89 MINN. L. REV. 917, 937–38 (2005) (observing that environmental resources suffer from a consumption problem (tragedy of the commons) while information resources suffer from a production problem (free rider dilemma)); Charles M. Schweik & Robert English, *Tragedy of the FOSS Commons? Investigating the Institutional Designs of Free/Libre and Open Source Software Projects*, FIRST MONDAY (2007), https://firstmonday.org/ojs/index.php/fm/article/view/1619/1534 [https://perma.cc/J23T-GXZ6] ("Free-riders in this context are programmers, testers or documenters who utilize a particular FOSS software but do not contribute back in these capacities. In a FOSS setting, the tragedy of the commons comes when there are insufficient human resources available to continue to further develop and maintain the software and, as a result, the software project is abandoned.").

148. *See* LESSIG, *supra* note 134, at 68–69 ("The problem instead is to assure a sufficient incentive to supply new or improved code — a provisioning problem . . . . Here, however, we

they argued, the real tragedy of the open-source commons would occur only if that precious cooperative ethos were ruined by the invasion of alien for-profit interests.[149]

Newer writers have called for more external help.[150] They have reframed the open-source commons as being not the code itself, but the attention of the developers who maintain the code.[151] That attention is finite and can be depleted, leading to neglect or abandonment of projects.[152] Unmaintained code deteriorates and depreciates rapidly with time.[153] Although it may seem efficient in the short term to use and build atop freely available code, that calculus ignores the negative

---

must work as empiricists, not ideologues. For we just have to look around to see the extraordinary amount of open code being written, despite the inability to control its copying."); Eric Raymond, *The Magic Cauldron* (1999), http://www.catb.org/esr/writings/magic-cauldron/magic-cauldron-5.html [https://perma.cc/SP2F-8HKQ] (stating that "it is empirically clear" that the breadth and volume of open-source development is increasing, so "there is some critical way in which the 'Tragedy of the Commons' model fails to capture what is actually going on"); *see also* Benkler, *supra* note 107, at 438–39 (arguing that information is nonrivalrous, so open source projects "can tolerate increasing levels of free-riding, as long as the absolute number of contributors . . . remains sufficiently large," though acknowledging that certain types of free-riding can adversely affect participation norms and quality control).

149. *See* LESSIG, *supra* note 134, at 23, 88, 175, 247 (characterizing the "tragedy of the innovation commons" as allowing each firm to increase its control over code without limit); Benkler, *supra* note 107, at 439–40 (commercialization would reduce incentives to participate); *see also* EGHBAL, *supra* note 144, at 59 ("Yet even within open source communities, there is a pervasive belief that money has a corrupting influence on open source.").

150. *See* Sharma, *supra* note 4, at 1193 ("Intervention is warranted when market failures are pronounced, their harms are intolerable, and they are unlikely to self-correct."); EGHBAL, *supra* note 19, at 157 ("[W]e don't yet have tidy answers to how online public goods are built, maintained, and paid for. . . . The need for a new approach is why Elinor Ostrom's writing has gained renewed appeal in recent years.").

151. *See* Sharma, *supra* note 4, at 1176 ("Open source is made up of two components — the code itself, which is inherently non-rivalrous, and the maintenance required to support it, which is not."); EGHBAL, *supra* note 19, at 161 ("Open source code, in static state, is a public good, meaning that it is both non-excludable and non-rivalrous. . . . The *production* of open source code, however, functions more like a commons — meaning that it is non-excludable and rivalrous — where *attention* is the rivalrous resource. Maintainers can't stop users from bidding for their attention, but their attention can be depleted."); *see also* WEBER, *supra* note 108, at 71 (reporting that the top twenty percent of programmers contribute eighty-one percent of open source code); Raymond, *supra* note 148 (noting that maintenance makes up the vast majority of what programmers do).

152. *See* Sharma, *supra* note 4, at 1182 ("By free-riding, Irresponsible Consumers are overusing the existing supply of open-source maintenance."); EGHBAL, *supra* note 144, at 78 ("[M]any open source projects are legacy tools, built once by a passionate developer or group of developers, who then lacked resources to manage their project's success. Over time, contributions decline as others get bored and move on, but the project is still in active use, leaving one or two people to figure out how to keep it alive."); *see also* SCHWEIK & ENGLISH, *supra* note 106, at 292–95 (finding that the most important factors distinguishing success from abandonment are "the number of hours worked on project," "a larger user community," and "elements of good leadership such as hard work, good planning, goal establishment and articulation, and project management"); Tim Wu, *Blind Spot: The Attention Economy and the Law*, 82 ANTITRUST L.J. 771, 783 (2019).

153. *See* EGHBAL, *supra* note 144, at 40 (contrasting physical infrastructure, which is "built to last," against digital infrastructure, which changes frequently and "requires frequent maintenance and upkeep to be compatible with other software components").

externalities of bad code.[154] In other words, open source as a commons is less like a lush field and more akin to a vast scrapyard. Although some treasures are well-maintained,[155] most of the dumped deposits are not. Best practices in secure coding now recommend de novo evaluation and recompilation of open-source code when integrating it into one's own software project,[156] which limits the direct efficiency gains of grazing at the commons.

Yet, even as the new guard expresses doubts about the efficiency of the open-source commons, they continue to see value in channeling resources and aid to prop up the status quo. Open source's productive potential may be less sumptuous than once imagined, but incumbency has a quality all its own.[157] Some open-source projects have become so ubiquitous that they might be considered critical infrastructure.[158] Economic estimates of replacement cost run into the billions or trillions of

154. *See* Sharma, *supra* note 4, at 1182; EGHBAL, ROADS AND BRIDGES, *supra* note 144, at 80–81 (noting that "[b]uilding digital infrastructure in a haphazard fashion means that all software gets built more slowly and inefficiently," and that "pure volunteer labor limit[s] the amount of security and reliability that [can] be provided to important software infrastructure"); Frank Nagle, *Open Source Software and Firm Productivity*, 65 MGMT. SCI. 1191, 1193–94 (2019) (acknowledging that long-term costs of using "free" software could be "5%-20% higher than those of proprietary closed source software" due to risks including "lack of development and support, security concerns, and lack of contractual relationships"); *see also* Chong, *supra* note 25 (explaining "technical debt" and critiquing risks of software "monocultures").

155. *See* WEBER, *supra* note 108, at 142 (noting that "open source generally has had more of an impact in settings like operating systems and less in end-user applications," because self-selecting developers want to work on challenging, complex projects that showcase their talent and artistry for a sophisticated audience).

156. *See* ENDURING SEC. FRAMEWORK, *supra* note 16, at 24; *Preventing Supply Chain Attacks Like SolarWinds*, LINUX FOUND. BLOG (Jan. 13, 2021), https://www.linuxfoundation.org/blog/blog/preventing-supply-chain-attacks-like-solarwinds [https://perma.cc/SWZ8-EZ27] (recommending the use of "verified reproducible builds" whereby "independent organizations produce a build from source code and verify that the built results come from the claimed source code").

157. *See* Sharma, *supra* note 4, at 1172–73 ("Given its benefits, the solution is not to move away from open source. In any event, it would be impractical, and perhaps impossible, to do so. It is too embedded in our systems, and the cost of replacing every open-source component with secure, newly developed closed-source code would be prohibitively expensive."). Other innovation systems are defended on strikingly similar epistemological grounds. *See* SUBCOMM. ON PATENTS, TRADEMARKS & COPYRIGHTS OF S. COMM. ON THE JUDICIARY, 85TH CONG., AN ECONOMIC REVIEW OF THE PATENT SYSTEM 80 (Comm. Print 1958) (prepared by Fritz Machlup) ("If we did not have a patent system, it would be irresponsible, on the basis of our present knowledge of its economic consequences, to recommend instituting one. But since we have had a patent system for a long time, it would be irresponsible on the basis of our present knowledge, to recommend abolishing it.").

158. *See* Office of the National Cyber Director, Request for Information on Open-Source Software Security: Areas of Long-Term Focus and Prioritization, 88 Fed. Reg. 54315 (2023) (stating that "open-source software plays a vital and ubiquitous role across the Federal Government and critical infrastructure"); Sharma, *supra* note 4, at 1167, 1172, 1205 ("Indeed, the open-source supply chain is arguably a National Critical Function itself.").

dollars.[159] Threats of liability could cause important open-source developers to withdraw their code and support from vital projects, thereby "breaking" the internet or other key software pipelines.[160] Thus, the dominant trend in the commentary has focused on carrots rather than sticks, out of fear that excessive deterrents could cause the open-source community to collapse and cascade toward critical failure.[161]

C. Free Expression

The third prevailing objection to open-source liability is the argument that software is speech. That assertion raises at least three layers of inquiry. The primary claim is that direct dissemination of code is an act of free expression that limits the power of government to regulate open-source software. A secondary question — given the dominant role of code-sharing sites in the open-source ecosystem — is whether intermediary platforms can assert independent speech protections of their own. If not, the residual question is whether individuals and platforms can collectively avoid legal oversight by invoking a free-associational right of anonymity.

Because software is composed of text, it necessarily has some expressive attributes.[162] Consequently, early U.S. cases acknowledged that publication of source code qualifies as "speech" covered by the First Amendment.[163] This stance has proved especially persuasive

---

159. *See* Manuel Hoffmann, Frank Nagle & Yanuo Zhou, *The Value of Open Source Software* 1 (Harv. Bus. Sch. Strategy Unit Working Paper, Paper No. 24-038, 2024) (estimating supply-side value of widely-used OSS to be $4.15 billion and the demand-side value to be $8.8 trillion, and that "firms would need to spend 3.5 times more on software than they currently do if OSS did not exist"); Carol Robbins, Gizem Korkmaz, Ledia Guci, José B. Santiago Calderón & Brandon L. Kramer, *A First Look at Open-Source Software Investment in the United States and in Other Countries, 2009-2019*, at 9 (2021), https://iariw.org/wp-content/uploads/2021/11/robbins-paper.pdf [https://perma.cc/RUV3-UH3B] (estimating "a capital stock value of open-source software of more than $118 billion dollars in 2019").

160. *Cf.* Sean Gallagher, *Rage-Quit: Coder Unpublished 17 Lines of Javascript and "Broke the Internet"*, ARS TECHNICA (Mar. 24, 2016, at 22:10 ET), https://arstechnica.com/information-technology/2016/03/rage-quit-coder-unpublished-17-lines-of-javascript-and-broke-the-internet/ [https://perma.cc/T6KR-4L35]. *But see* SCHWEIK & ENGLISH, *supra* note 106, at 34 (observing that open-source foundations have "formed to protect developers involved in specific projects, especially against potential lawsuits").

161. *See* EGHBAL, *supra* note 144, at 47.

162. *See* NAT'L COMM'N ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REPORT 15 (1978) (observing that computer programs "are writings in the constitutional sense and eligible for copyright if Congress so provides"). *But see id.* at 28 (dissenting that computer programs are not writings because they "are addressed to machines"); Pamela Samuelson, *CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine-Readable Form*, 1984 DUKE L.J. 663, 663 (observing that "computer programs in machine-readable form do not disclose their contents and are inherently utilitarian").

163. *See* Bernstein v. U.S. Dep't of Justice, 176 F.3d 1132, 1142 (9th Cir. 1999), *reh'g granted en banc and opinion withdrawn*, 192 F.3d 1308 (9th Cir. 1999); Junger v. Daley, 209

when code is presented as, or in conjunction with, scientific research. At the extreme, any act of free software publication could be framed as expressing a political ideology rooted in autonomy and open information exchange.[164]

Yet, code also has overwhelming "nonspeech" functional attributes.[165] The pivotal question has turned out to be not whether, but to what extent, code is constitutionally protected.[166] Most courts to consider the question have concluded that First Amendment protection for software is quite thin.[167] By extrapolation, when software execution causes cognizable harm, free expression principles are unlikely to preempt tort claims against the developers and maintainers of that software.[168]

The more interesting question is whether free-speech principles should shield code-hosting platforms like GitHub from intermediary

---

F.3d 481, 485 (6th Cir. 2000); Universal Studios, Inc. v. Corley, 273 F.3d 429, 446 (2d Cir. 2001).

164. *See* RAYMOND, *supra* note 109, at 194 (describing open source as "a broadly libertarian view of the proper relationship between individuals and institutions"); STALLMAN, *supra* note 108. *But see* United States v. O'Brien, 391 U.S. 367, 376 (1968) ("We cannot accept the view that an apparently limitless variety of conduct can be labeled 'speech' whenever the person engaging in the conduct intends thereby to express an idea."); *cf.* EGHBAL, *supra* note 19, at 33 ("The simple act of using GitHub already separates [today's] developers from the most dogmatic free software advocates, who refuse to use the platform whatsoever. . . . The GitHub generation of open source developers doesn't feel particularly strongly about these issues. They just want to make things, and sharing is a natural byproduct of those efforts.").

165. *See* Pamela Samuelson, Randall Davis, Mitchell D. Kapor & J. H. Reichman, *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2319 (1994) ("Programs have almost no value to users as texts. Rather, their value lies in behavior.").

166. *See* United States v. Osadzinski, 97 F.4th 484, 491–92 (7th Cir. 2024); Green v. U.S. Dep't of Justice, 54 F.4th 738, 745 (D.C. Cir. 2022); Corley, 273 F.3d at 452 ("The functionality of computer code properly affects the scope of its First Amendment protection."); Tim Wu, *Machine Speech*, 161 U. PA. L. REV. 1495, 1520 (2013); Dan L. Burk, *Software as Speech*, 8 SETON HALL CONST. L.J. 683, 685–86 (1998).

167. *See* Kyle Langvardt, *Crypto's First Amendment Hustle*, 26 YALE J.L. & TECH. 130, 152 (2023) ("[W]hen the interference with expression is merely incidental — as it has been in every case except *Bernstein* — courts apply intermediate scrutiny. The government has never once failed to clear the bar in one of these cases. As a result, courts applying the allegedly formidable 'code is speech' principle have repeatedly upheld schemes that impose heavy, even excessive regulatory penalties on software developers.").

168. *See* Ryan M. Calo, *Open Robotics*, 70 MD. L. REV. 571, 599 (2011); Singh v. Edwards Lifesciences Corp., 210 P.3d 337, 342–43 (Wash. Ct. App. 2009); Ramirez v. Paradies Shops, LLC, 69 F.4th 1213, 1220 (11th Cir. 2023) (finding that defendant "is not shielded from liability by the intervening criminal act of the cybercriminals" because it "could have foreseen being the target of a cyberattack"). *See generally* Kenneth S. Abraham & G. Edward White, *First Amendment Imperialism and the Constitutionalization of Tort Liability*, 98 TEX. L. REV. 813, 844 (2020) (arguing that First Amendment "imperialism" should not extend to tort liability); Alfred C. Yen, *Rethinking Copyright's Relationship to the First Amendment*, 100 B.U. L. REV. 1215, 1217 n.1 (2020) (collecting sources on the "orthodoxy" that First Amendment principles are subsumed within copyright doctrine). *But see* David A. Anderson, *First Amendment Limitations on Tort Law*, 69 BROOK. L. REV. 755, 782 (2004) (observing that "courts have universally assumed that tort liability is neither exempted from nor foreclosed by the First Amendment").

liability and other regulatory constraints.[169] A foundational axiom of cyberlaw is that harnessing technological intermediaries as points of control will suppress protected speech activities by individual users of those technologies.[170] In deference to that axiom, courts have wielded Section 230 of the Communications Decency Act to shortcut the more difficult First Amendment analysis.[171] Nevertheless, multiple generations of cyberlaw scholars have noted that faithful application of First Amendment first principles would likely yield a different equilibrium.[172]

For nearly three decades, Section 230 provided a virtually impenetrable shield for internet platforms against liability claims relating to third-party speech content.[173] Until recently, any effort to fault such platforms for the provision (or takedown) of third-party speech content

169. *See* Hon. John G. Browning, *A Product By Any Other Name? The Evolving Trend of Product Liability Exposure for Technology Platforms*, 16 ELON L. REV. 181, 182–83 (2024); *see also* Matthew B. Lawrence, *Public Health Law's Digital Frontier: Addictive Design, Section 230, and the Freedom of Speech*, 4 J. FREE SPEECH L. 299, 340–42 (2024) (promoting a "neutrality triangulation approach" that would allow government to regulate content-neutral conduct by a platform vis-à-vis users); Evelyn Douek, *Content Moderation as Systems Thinking*, 136 HARV. L. REV. 526, 584–86 (2022) (advocating a "second-wave content moderation regulatory model" that brings a systems thinking approach to procedural decisions that reside upstream from resolution of individual cases).

170. *See* Jonathan Zittrain, *Internet Points of Control*, 43 B.C. L. REV. 1 (2003); Christina Mulligan, *Technological Intermediaries and Freedom of the Press*, 66 S.M.U. L. REV. 157 (2013); Margot E. Kaminski & Shane Witnov, *The Conforming Effect: First Amendment Implications of Surveillance, Beyond Chilling Speech*, 49 U. RICH. L. REV. 465, 499 (2015); Eric Goldman, *Why Section 230 Is Better Than the First Amendment*, 95 NOTRE DAME L. REV. REFLECTIONS 33, 36–39 (2019). *See generally* Grimmelmann & Zhang, *supra* note 75 (collecting literature).

171. *See* Blake E. Reid, *Section 230's Debts*, 22 FIRST AMEND. L. REV. 408, 416–17 (2024) (arguing that First Amendment doctrine is underdeveloped because of § 230); Danielle Keats Citron & Mary Anne Franks, *The Internet As a Speech Machine and Other Myths Confounding Section 230 Reform*, 2020 U. CHI. LEGAL F. 45, 59–60 (2020) (arguing that "the conflation of Section 230 and the First Amendment short-circuits" analysis of whether online activity is speech and whether it should be protected).

172. See Wu, *supra* note 75, at 340 (suggesting collateral censorship is not a problem when the intermediary is the direct target of liability); Jeff Kosseff, *First Amendment Protection for Online Platforms*, 35 COMPUT. L. & SEC. REV. 199 (2019) (arguing that First Amendment doctrine offers limited protections to online platforms in the absence of § 230); Rebecca Tushnet, *Power Without Responsibility: Intermediaries and the First Amendment*, 76 GEO. WASH. L. REV. 986, 1004 (2008) (noting that "the CDA's policy of conferring complete immunity on ISPs is not inevitable and, most significantly, not currently understood as a First Amendment requirement"); Kyle Langvardt & Alan Z. Rozenshtein, *Beyond the Editorial Analogy: First Amendment Protections for Platform Content Moderation After* Moody v. NetChoice, 6 J. FREE SPEECH L. 1, 28 (2025) (observing that the Supreme Court's recent pronouncement on platform governance "does not automatically call for strong First Amendment protection"); Tim Wu, *Is the First Amendment Obsolete?*, 117 MICH. L. REV. 547, 571 (2018) (arguing that the First Amendment should impose affirmative duties on platforms); *cf.* Balkin, *supra* note 74, at 1229–30 (predicting that First Amendment conflicts will tend to favor internet platforms over end users, and that internet reformers then will seek to sideline First Amendment inquiries in order to regulate internet platforms).

173. *See* Jeff Kosseff, *A User's Guide to Section 230, and a Legislator's Guide to Amending It (Or Not)*, 37 BERKELEY TECH L.J. 757, 773–79 (2022).

was flatly denied. The only notable exceptions were intellectual property claims and criminal charges (by statute),[174] instances where the platform actively participated in co-creating the content,[175] and claims entirely unrelated to the platform's role as publisher or speaker of third-party content.[176]

Yet, as political esteem for Section 230 has waned, judicial consensus has followed suit.[177] In an emerging trend, courts have held that a growing number of "content-neutral" software design choices fall outside Section 230's scope.[178] The two lead cases involved Snapchat, a social media app, and the design of a "speed filter" feature that encouraged reckless driving resulting in multiple deaths.[179] The plaintiffs' central claim was that the dangerousness of the software stemmed solely from antecedent design and architecture decisions, and had nothing to do with post hoc "editing, monitoring, or removing" of content generated by third-party users.[180] Surprisingly, the courts agreed, holding that Section 230 does not apply to claims of negligent or defective software design. And because the speed filter feature was embedded

---

174. 47 U.S.C. § 230(e).

175. *See* Kosseff, *supra* note 173, at 780 (discussing Fair Hous. Council of San Fernando Valley v. Roommates.com, LLC, 521 F.3d 1157, 1166 (9th Cir. 2008) (en banc)); Liapes v. Facebook, Inc., 313 Cal. Rptr. 3d 330, 345 (Cal. Ct. App. 2023); *see also* Lawrence, *supra* note 169, at 339–40 (explaining the *Roommates.com* holding).

176. *See* Kosseff, *supra* note 173, at 783 (discussing Barnes v. Yahoo! Inc., 570 F.3d 1096 (9th Cir. 2009)).

177. *See* Citron & Franks, *supra* note 171, at 46 (noting that "politicians across the ideological spectrum are raising concerns about the leeway provided to content platforms under Section 230").

178. *See* Lawrence, *supra* note 169, at 332, 340 (explaining that the "'content-neutrality' test focuses on whether a cause of action or claim regulates platform conduct that is itself content neutral as between the platform and users"); Browning, *supra* note 169, at 203 (observing that "a crucial difference exists" in "those lawsuits that target an app's design function itself as the defect"); *In re* Zoom Video Commc'ns, Inc. Priv. Litig., 525 F. Supp. 3d 1017, 1034 (N.D. Cal. 2021) (holding that § 230 "immunizes liability deriving from moderation of third-party content" but "allows claims that . . . are content-neutral"); *see also* Olivier Sylvain, *Intermediary Design Duties*, 50 CONN. L. REV. 203, 258–59 (2018) (predicting that courts "may come to recognize that service providers' design of the choice architecture precipitate illegal expressive acts"); Andrea M. Matwyshyn, *Hidden Engines of Destruction: The Reasonable Expectation of Code Safety and the Duty to Warn in Digital Products*, 62 FLA. L. REV. 109, 146 (2010) (arguing that a tort-based duty of reasonable code safety is a content-neutral regulation that passes First Amendment scrutiny); Bernstein, 176 F.3d at 1145 (suggesting in dicta that software "products" could be regulated via "content-neutral time, place and manner restrictions that may have an incidental effect on expression while aiming at secondary effects"). *But see* Balkin, *supra* note 74, at 1238 ("A platform does not even have to be primarily for exchanging ideas . . . . Hence Evelyn Douek's aphorism that '[e]verything is content moderation.'").

179. *See* Lemmon v. Snap, Inc., 995 F.3d 1085, 1088 (9th Cir. 2021); Maynard v. Snapchat, Inc., 870 S.E.2d 739, 743 (Ga. 2022); *see also* Browning, *supra* note 169, at 204–06 (discussing *Lemmon* as the "first case to expose a chink in the armor of tech developers' reliance on Section 230 to deflect product liability claims").

180. *Lemmon*, 995 F.3d at 1092–93 ("[T]he Parents' negligent design claim faults Snap solely for Snapchat's architecture, contending that the app's Speed Filter and reward system worked together to encourage users to drive at dangerous speeds.").

directly into user-created videos, the Snapchat cases set a new precedent that Section 230 does not bar claims that are merely "content-related."[181]

Subsequent cases have begun testing the bounds of this new "content-neutral design" rule, creating a new zone of uncertainty. On one end, some courts have held claims to be content-neutral when addressing cybersecurity failures,[182] noncompliance with offline regulations,[183] or addictive features such as continuous scrolling, rewards for engagement, and sock accounts.[184] Meanwhile, courts have split on whether algorithmic "matchmaking" features that match criminals with victims should be treated as content-based[185] or content-neutral.[186] At the other end, courts have largely sustained that Section 230 still applies to identity verification or age verification requirements, and have rejected efforts to characterize them as content-neutral security

181. *See* Lawrence, *supra* note 169, at 329 ("*Lemmon* does seem to vitiate the argument that a claim's connection to content automatically brings it within the scope of Section 230. Content — the content submitted by the users — was inextricable (and indispensable) to the claims in the case.").

182. *See In re Zoom*, 525 F. Supp. 3d at 1030 (stating that "the text of § 230(c) immunizes the 'blocking and screening of offensive material,' not failures to secure software from intrusion").

183. *See* HomeAway.com, Inc. v. City of Santa Monica, 918 F.3d 676, 679–80 (9th Cir. 2019) (failure to check compliance with home-share licensing laws); Nunes v. Twitter, Inc., 194 F. Supp. 3d 959, 960 (N.D. Cal. 2016) (delivering unwanted messages via text to plaintiff's phone); Lee v. Amazon.com, Inc., 291 Cal. Rptr. 3d 332, 338 (Cal. Ct. App. 2022) (failure to warn of products containing mercury).

184. *See In re* Coordinated Proc. Special Title Rule 3.550 Soc. Media Cases, No. 22STCV21355, 2023 Cal. Super. LEXIS 76992, at *96–98 (Cal. Super. Ct. Oct. 13, 2023) ("As in *Lemmon*, Plaintiffs' claims based on the interactive operational features of Defendants' platforms do not seek to require that Defendants publish or de-publish third-party content that is posted on those platforms. The features themselves allegedly operate to addict and harm minor users of the platforms regardless of the particular third-party content viewed by the minor user."); *In re* Soc. Media Adolescent Addiction/Personal Inj. Prods. Liab. Litig., 702 F. Supp. 3d 809, 829 (N.D. Cal. 2023) (allowing defect allegations including lack of effective parental controls; inability to self-restrict time used on a platform; difficulty deleting accounts; lack of robust age verification; difficulty reporting predator accounts; offering appearance-altering filters; not labeling filtered content; and timing of notification alerts to increase addictive use).

185. *See* Dyroff v. Ultimate Software Grp., Inc., 934 F.3d 1093, 1094–96 (9th Cir. 2019); Herrick v. Grindr LLC, 765 F. Appx. 586, 588–89 (2d Cir. 2019); Doe v. Backpage.com, LLC, 817 F.3d 12, 15–17 (1st Cir. 2016).

186. *See* A.M. v. Omegle.com, LLC, 614 F. Supp. 3d 814, 817 (D. Or. 2022); Doe v. Internet Brands, Inc., 824 F.3d 846, 848 (9th Cir. 2016); *see also* Lawrence, *supra* note 169, at 322 ("There is at this writing a significant, unresolved legal controversy about the extent to which Section 230(c)(1) bars regulation of website conduct to prioritize, recommend, or otherwise steer users toward particular content."). *Compare* Moody v. NetChoice, LLC, 603 U.S. 707, 734 (2024) (stating in dicta that "prioritization of content, achieved through the use of algorithms" is expressive activity protected by the First Amendment because it exercises an "editorial function" of "compiling and curating others' speech"), *with id.* at 745–46 (Barrett, J., concurring) (positing that automated algorithmic actions are less expressive than human-based review), *and id.* at 728 (Alito, J., concurring) (distinguishing a "curator" from a "dumb pipe").

features.[187] To be clear, most legal complaints brought against platforms remain squarely in the realm of content moderation, and courts consistently reject artful efforts to plead around that fact.[188]

But not enough cases have been decided yet to draw conclusive lines. To take just the last example, identity verification measures are surely problematic when applied broadly to general-purpose websites and social media platforms, because of conformity effects and collateral censorship.[189] At the same time, as the Supreme Court recently held, identification requirements can serve a functional, content-neutral purpose in more targeted contexts such as driver's licenses, alcohol-serving establishments, pharmaceutical sales, and obscenity.[190] That content neutrality is heightened where the primary regulatory focus is on conduct rather than on expression.[191] Whether a code-hosting

187. *See* Bride v. Snap, Inc., No. 21-cv-06680, 2023 WL 2016927, at *1–2, *8 (C.D. Cal. Jan 10, 2023); Doe v. Snap, Inc., No. H-22-00590, 2022 WL 2528615, at *12–13 (S.D. Tex. July 7, 2022); Doe v. Grindr Inc., No. 2:23-cv-02093, 2023 WL 9066310, at *1050–51 (C.D. Cal. Dec. 28, 2023); Doe v. Grindr, LLC, No. 5:23-cv-193, 2023 WL 7053471, at *1–3 (M.D. Fla. Oct. 26, 2023); *see also* Doe v. Myspace, Inc., 528 F.3d 413, 415–18 (5th Cir. 2008) (holding that a negligence claim alleging a lack of sufficient age verification measures is barred by § 230). *But see In re* Soc. Media Adolescent Addiction/Personal Inj. Prods. Liab. Litig., 702 F. Supp. 3d at 829–30 (finding that age verification claims are not barred by § 230 because they are broader in ways that would not impact publication of third-party content). Additionally, multiple courts have struck down a wave of age verification statutes as unconstitutional under the First Amendment. *See* Free Speech Coalition, Inc. v. Rokita, 738 F. Supp. 3d 1041, 1048–49, 1069 (S.D. Ind. 2024); NetChoice, LLC v. Fitch, 738 F. Supp. 3d 753, 764, 780 (S.D. Miss. 2024); NetChoice, LLC v. Yost, 716 F. Supp. 3d 539, 546, 561 (S.D. Ohio 2024); NetChoice, LLC v. Griffin, No. 5:23-cv-05105, 2023 WL 5660155, at *1, *21 (W.D. Ark. Aug. 31, 2023). *But see* Free Speech Coalition, Inc. v. Paxton, 95 F.4th 263, 267 (5th Cir. 2024) (defying the trend and applying rational basis review to uphold statute), *aff'd on other grounds*, 606 U.S. 461 (2025) (applying intermediate scrutiny).

188. *See* Browning, *supra* note 169, at 211 (observing that "*Lemmon* and *Maynard* have not exactly ushered in a wave of pro-plaintiff results for litigants asserting product liability claims against tech companies").

189. *See* Reno v. ACLU, 521 U.S. 844, 849, 880 (1997); Kaminski & Witnov, *supra* note 170; Wu, *supra* note 75, at 293; Julie E. Cohen, *A Right to Read Anonymously: A Closer Look at "Copyright Management" in Cyberspace*, 28 CONN. L. REV. 981, 1003–04 (1996).

190. *See Paxton*, 606 U.S. at 479 ("Requiring age verification is common when a law draws lines based on age."); Branscomb, *supra* note 124, at 1643 n.10 ("Drivers on the real highways are not permitted access without a motor vehicle license so, logically, it should be possible to require users of networks to be responsible and identifiable."); Council for Responsible Nutrition v. James, No. 24-cv-1881, 2024 WL 1700036, at *6 (S.D.N.Y. Apr. 19, 2024) (dismissing First Amendment challenge of age restriction statute for the purchase of dietary supplements); *see also* United States v. O'Brien, 391 U.S. 367, 381–82 (1968) (holding that statutory prohibition of knowing destruction or mutilation of Selective Service registration certificates was content-neutral and supported by substantial state interest); *cf.* Eugene Volokh, *The Law of Pseudonymous Litigation*, 73 HASTINGS L.J. 1353, 1367 (2022) (explaining the general presumption against anonymous litigation).

191. *See Paxton*, 606 U.S. at 516 (2025) (Kagan, J., dissenting) (stating that the Court has applied intermediate scrutiny to laws where "the regulation is of conduct, and the burden on expression a rare knock-on effect"); KENT GREENAWALT, FIGHTING WORDS 6 (1995) (stating that the reasons for protecting free speech do not extend to "situation-altering" communications that "dominantly represent commitments to action, not assertions of facts or values or

platform that facilitates admixture and distribution of anonymous executable code is more analogous to an interactive bulletin board or to the provision of regulated substances remains an unsettled question. But if the latter, it seems unlikely that Section 230 will provide safe harbor against otherwise-valid identity verification requirements.

Setting aside questions of code-as-speech and Section 230 immunity, the question of identity verification raises separate issues of free association and the so-called right to anonymity. Although the Supreme Court has declared at times that the First Amendment protects a "right to anonymity,"[192] the doctrine has proved much more limited in application.[193]

Typically, the cases most strongly in support entail a likelihood of threats, harassment, or reprisals against peaceable and sympathetic groups.[194] Other groups perceived as violent or dangerous do not receive the same benefit of the doubt.[195] The manner of disclosure also matters: face-to-face identification requirements tend to be allowed unless there is a credible threat of imminent retaliation.[196] Advance registration requirements are more delicate: for ordinary activities, requiring preapproval from the government is disfavored because it impinges on spontaneous speech and voluntary access to constitutionally protected

---

expressions of feeling"). *But see* Eugene Volokh, *Crime-Facilitating Speech*, 57 STAN. L. REV. 1095, 1217 (2005) (arguing in favor of only limited First Amendment exception for crime-facilitating speech).

192. *See* McIntyre v. Ohio Elections Comm'n, 514 U.S. 334, 357 (1995); Talley v. California, 362 U.S. 60, 64–65 (1960).

193. *See* Lyrissa Barnett Lidsky & Thomas F. Cotter, *Authorship, Audiences, and Anonymous Speech*, 82 NOTRE DAME L. REV. 1537, 1538 (2007); Nathan J. Ristuccia, *"Dangerous to the Liberties of a Free People": Secret Societies and the Right to Assemble*, 4 J. FREE SPEECH L. 139, 143–45 (2023); Kaminski, *supra* note 123, at 850 ("The variation in anti-mask statutes suggests that legislatures, like courts, struggle with determining when anonymity is functional and when it is expressive."). *See generally* Choi, *supra* note 124 (collecting sources).

194. *See* Ams. for Prosperity Found. v. Bonta, 594 U.S. 595, 616–18 (2021); Watchtower Bible & Tract Soc'y of N.Y., Inc. v. Village of Stratton, 536 U.S. 150, 164 (2002); Bates v. City of Little Rock, 361 U.S. 516, 524 (1960); NAACP v. Alabama *ex rel.* Patterson, 357 U.S. 449, 459–60 (1958).

195. *See* Communist Party of the U.S. v. Subversive Activities Control Bd., 367 U.S. 1, 88–89 (1961); Barenblatt v. United States, 360 U.S. 109, 126 (1959); New York *ex rel.* Bryant v. Zimmerman, 278 U.S. 63, 71–72 (1928); Church of the Am. Knights of the Ku Klux Klan v. Kerik, 356 F.3d 197, 208–09 (2d Cir. 2004).

196. *Compare* Hiibel v. Sixth Jud. Dist. Ct. of Nev., 542 U.S. 177, 191 (2004) ("One's identity is, by definition, unique; yet it is, in another sense, a universal characteristic. Answering a request to disclose a name is likely to be so insignificant in the scheme of things as to be incriminating only in unusual circumstances."), *and* Doe v. Reed, 561 U.S. 186, 200–01 (2010) (refusing to strike down statute requiring disclosure of signatories to referendum petitions where the record shows "scant evidence" of "threats, harassment, or reprisals"), *with* Buckley v. Am. Const. L. Found., Inc., 525 U.S. 182, 198–200 (1999) (noting that petition circulators experience harassment and retaliation, and concluding that the statute requiring petition circulators to wear identification badges "compels personal name identification at the precise moment when the circulator's interest in anonymity is greatest").

materials.[197] By contrast, for regulated activities, registration requirements are typically held valid,[198] unless the government's stated interest is found to be pretextual or nonexistent.[199] What constitutes a "regulated" activity is often just a matter of governmental say-so.[200]

While open-source maintainers frequently report being subject to harassment and abuse by users of and contributors to their projects,[201] this type of generalized risk of having an online presence is unlikely to be analogous to the risk of imminent, physical violence described by canonical right-to-anonymity cases. Instead, the more plausible claim is that the regulatory burden of advance registration requirements is likely to suppress spontaneous speech or other protected speech activities.

Arguably, the Supreme Court's recent decision in *Americans for Prosperity Foundation v. Bonta*[202] offers support for the latter claim. The issue in *Bonta* was whether California violated the First Amendment right to free association by requiring all charitable organizations to disclose the identities of their major donors. The Court applied

197. *See Watchtower*, 536 U.S. at 166–67; *see also* Reno v. ACLU, 521 U.S. 844, 855–56, 881–82 (1997) (finding that adult registration requirements to view online materials would impose an unacceptably heavy burden on constitutionally protected speech); Ashcroft v. ACLU, 542 U.S. 656, 666–67 (2004) (finding that adult registration fails to be the least restrictive means of restricting children's access to online materials harmful to them).

198. *See* McConnell v. Fed. Election Comm'n, 540 U.S. 93, 199, 201 (2003) (upholding disclosure requirements on political contributions and expenditures where there was a "lack of specific evidence" that anyone would be prevented from speaking); Buckley v. Valeo, 424 U.S. 1, 69–70 (1976) (stating that "any serious infringement on First Amendment rights brought about by the compelled disclosure of [political] contributors is highly speculative"); United States v. Harriss, 347 U.S. 612, 625 (1954) (stating that registration requirements for lobbyists "merely provided for a modicum of information from those who for hire attempt to influence legislation or who collect or spend funds for that purpose"); Connection Distrib. Co. v. Holder, 557 F. 3d 321, 325, 328 (6th Cir. 2009) (upholding age verification requirements for producers and performers of sexually explicit materials); *cf.* Heller v. District of Columbia, 801 F.3d 264, 274 (D.C. Cir. 2015) (upholding basic registration requirement for gun owners). *But see* McConnell, 540 U.S. at 275–76 (Thomas, J., dissenting in part) ("[T]his Court has explicitly recognized that 'the interest in having anonymous works enter the marketplace of ideas unquestionably outweighs any public interest in requiring disclosure as a condition of entry.'" (quoting McIntyre v. Ohio Elections Comm'n, 514 U.S. 334, 342 (1995))).

199. *See* Shelton v. Tucker, 364 U.S. 479, 488 (1960) (invalidating Arkansas statute requiring teachers to disclose all organizational memberships, many of which "could have no possible bearing upon the teacher's occupational competence or fitness"); Thomas v. Collins, 323 U.S. 516, 526, 540 (1945) (overturning registration requirement for labor organizers, where Texas refused to identify any governmental interest beyond routine regulation of business practice); Ams. for Prosperity Found. v. Bonta, 594 U.S. 595, 614 (2021) (striking down a state's "blanket demand" for disclosure of all charities' top donors as a "dragnet for sensitive donor information").

200. *Cf.* Bryan H. Choi, *For Whom the Data Tolls: A Reunified Theory of Fourth and Fifth Amendment Jurisprudence*, 37 CARDOZO L. REV. 185, 201–03 (2015) (observing that the mere legislative act of requiring records to be kept can be enough to turn an ordinary activity into a regulated one (citing *In re* M.H., 648 F.3d 1067, 1074 (9th Cir. 2011))).

201. *See, e.g.*, Lawson, *supra* note 12.

202. 594 U.S. 595 (2021).

exacting scrutiny and struck down the statute for being likely to chill donor contributions.[203] Although the Court acknowledged California has a substantial interest in preventing fraud, it reasoned that the disclosure requirement "casts a dragnet for sensitive donor information" without demonstrating even "a single concrete instance in which [such collection] did anything to advance the Attorney General's investigative, regulatory or enforcement efforts."[204]

Registration requirements for open-source contributions will surely deter or chill some open-source participants, especially those who identify strongly with the hacker ethic. If open-source code contributions resemble a form of charitable donation, then perhaps the *Bonta* decision can be extended to code-hosting platforms as recipients of charitable contributions. In other words, compelling code-hosting platforms to disclose the identities of their members infringes on those members' free-association rights.

But *Bonta* can also be distinguished on the facts. Code is not cash. The national security and consumer safety implications of open-source software provide more compelling governmental interests in regulating code-hosting platforms than the anti-fraud justifications in regulating charities.[205] Moreover, there is substantial evidence that identification would be actually effective in deterring code poisoning attacks and in improving other code safety concerns.[206] And unlike the entities in *Bonta*, GitHub and other popular code-hosting platforms are privately owned and are not themselves charitable institutions subject to public oversight.[207]

In sum, free expression principles have important applicability in considering open-source liability. But their significance has been narrowly construed in the code-as-speech context, particularly where the functional aspects dominate the expressive aspects. By sharp contrast, internet intermediaries have long enjoyed broad protections against pass-through liability for end-user activities. Still, those protections are beginning to erode. Code-hosting platforms present the type of heightened risks that could justify special exception to the usual rule of intermediary immunity.

---

203. *Id.* at 616–18.

204. *Id.* at 613.

205. *See* THE WHITE HOUSE, *supra* note 7.

206. *See supra* notes 66 & 125 (documenting the need for multifactor authentication and other identity measures).

207. *See* C. Edwin Baker, *First Amendment Limits on Copyright*, 55 VAND. L. REV. 891, 898 (2002) (noting that "the press is typically a corporate entity" and that "[i]ntelligent media policy often requires the government to intervene appropriately").

IV. THE LAW OF TAINTED DONATIONS

Conventional treatments of software liability assume as axiomatic that it will be too difficult or undesirable to impose liability on open-source software developers and platforms.[208] Yet, as a general rule, there is no special exemption from tort duties for free or voluntary actions.

An overview of the law of tainted donations offers several relevant lessons for software law. The broad takeaway is that general tort principles provide no special exemption for charitable acts of donation. Even where legislatures have enacted shield laws, they typically leave in place negligence-based duties of care. This rule extends to "unavoidably unsafe" products. Even when certain risks are inherent and cannot be detected or eliminated in advance, donors are expected to exercise due care where feasible.

In selecting a liability rule, the importance of the donation activity is often discussed at length but does not seem to negate the need for a negligence-like framework. Fears of deterring future donors fail to override competing concerns of safety and victim redress. When harm to innovation is raised, it is more often invoked as a critique of safe harbors that reduce financial incentives to develop and adopt safer techniques, rather than as a need for donors to subsidize research and commercialization.

Those lessons map closely onto the first two objections discussed above against imposing legal accountability on open-source contributors.

A. Food Donations

Tainted food holds a special place in the tort canon. It is well-established that much of the modern tort law regime evolved out of early case law on adulterated food provisions.[209] Courts had long recognized that sales of food carry a special implied warranty of wholesomeness

208. *See* HAMIN ET AL., *supra* note 6, at 13–15 (surveying literature and finding that "none endorsed the idea of including developers of [open-source software] in a software liability regime"). *But see* Meyers & Gilbert, *supra* note 6 ("To improve cybersecurity, open source software should not be completely exempt from software liability.").

209. *See* DAVID G. OWEN, PRODUCTS LIABILITY LAW 461–62 (3d ed. 2015); Denis W. Stearns, *Prosser's Bait-and-Switch: How Food Safety Was Sacrificed in the Battle for Tort's Empire*, 15 NEV. L.J. 106, 108 (2014); *see also* Alexandra D. Lahav, *A Revisionist History of Products Liability*, 122 MICH. L. REV. 509, 557 (2023) (describing common law evolution from "medicines in the 1850s and 1860s, then canned meat, clothing, furnishings, and finally machines").

or fitness for consumption.[210] But prior to the twentieth century, courts had limited such warranties according to the rule of privity, thus precluding recovery against entities further upstream in the supply chain.[211]

As industrial manufacturing of food products accelerated at the turn of the twentieth century, muckraking journalism and advocacy led by Dr. Harvey Wiley generated public pressure culminating in the enactment of the Pure Food and Drugs Act of 1906.[212] Accompanying that groundswell, judicial courts felt increasingly empowered to discard the privity defense, specifically in cases involving food and other "imminently dangerous" articles.[213] Concomitantly, courts converted the contract-based warranty of purity into a negligence-based duty of care.[214]

Food law continued to feature prominently during the drafting of Section 402A of the Second Restatement of Torts, which famously set off the strict products liability revolution. When the first draft of Section 402A was adopted in 1961, the scope of the rule was limited to sales of "*food* in a defective condition."[215] In subsequent drafts, Prosser and the American Law Institute ("ALI") successfully broadened the proposed rule first to "products for intimate bodily use" and finally to

---

210. *See* OWEN, *supra* note 209, at 248; Emerson v. Brigham, 10 Mass. 197, 201, 203 (1813) (noting that "the very offer to sell [food provisions at a sound price] is a representation or affirmation of the soundness of the article," but finding no liability where sellers had no knowledge of food spoilage at the time of sale). *But see* Farrell v. Manhattan Market Co., 84 N.E. 481, 487 (Mass. 1908) (finding no implied warranty where food was sold at the bargain counter and the selection was made by the buyer).

211. *See* OWEN, *supra* note 209, at 183–84; William G. Prosser, *The Implied Warranty of Merchantable Quality*, 27 MINN. L. REV. 117, 119 (1943).

212. *See* Pub. L. No. 59-384, 34 Stat. 768, *superseded by* Federal Food, Drug, and Cosmetic Act of 1938, Pub. L. No. 75-717, 52 Stat. 1040 (establishing the FDA); Fred B. Linton, *Federal Food and Drug Laws — Leaders Who Achieved Their Enactment and Enforcement*, 50 FOOD & DRUG L.J. 9, 14 (1995).

213. *See* OWEN, *supra* note 209, at 184 n.296 (collecting early cases involving defective foodstuffs); Tomlinson v. Armour & Co., 70 A. 314, 317 (N.J. 1908) (holding that the manufacturer of canned meats owes a duty to "the ultimate consumer to exercise care that the goods which he puts into cans and sells to retail dealers . . . are wholesome and fit for food, and not tainted with poison"); Watson v. Augusta Brewing Co., 52 S.E. 152, 152–53 (Ga. 1905) (finding that a bottler owes a duty to the general public to avoid mixing broken glass in a beverage, and that it does not matter that there was no privity of relationship); Salmon v. Libby, McNeil & Libby, 76 N.E. 573, 573–74 (Ill. 1905) (manufacturer owes duty not to sell poisonous canned mincemeat); *see also* Huset v. J.I. Case Threshing Mach. Co., 120 F. 865, 870 (8th Cir. 1903) (grouping poisonous foods with other "imminently dangerous" articles).

214. *See, e.g.*, Boyd v. Coca Cola Bottling Works, 177 S.W. 80, 81 (Tenn. 1914); Ketterer v. Armour & Co., 200 F. 322, 323 (S.D.N.Y. 1912).

215. *See* Stearns, *supra* note 209, at 108–09 (quoting RESTATEMENT (SECOND) OF TORTS § 402A (A.L.I., Tentative Draft No. 6, 1961)) (noting that "a large majority" of voting members approved the application of the rule to food manufacturers but that they were "about evenly divided" on extending it to other retailers and wholesalers).

"any product."[216] Notwithstanding those expansionist efforts, food continues to remain the paradigmatic case for strict products liability.[217] Harm caused by defective or tainted food can provide a cause of action against any or all commercial entities (e.g., manufacturers, wholesalers, or retailers) who have participated in the food supply chain.[218]

Donated food items did not receive special exemption at common law.[219] As a general rule, even good Samaritans owe a common duty of care when intervening to help those in need.[220] That said, governmental food relief actions were generally shielded by sovereign immunity.[221] And private institutional food donation activities were likely to be covered by the charitable immunity doctrine.[222] First adopted in 1876 by

---

216. *Compare* RESTATEMENT (SECOND) OF TORTS § 402A (A.L.I., Tentative Draft No. 7, 1962), *with* RESTATEMENT (SECOND) OF TORTS § 402A (A.L.I., Tentative Draft No. 10, 1964). *See also* Putnam v. Erie City Mfg. Co., 338 F.2d 911, 918–19 (5th Cir. 1964).

217. *See* RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 7 (A.L.I. 1998) (explaining that consumer expectations are "sufficiently well-formed" to supply a standard of defect in the food context, unlike in other contexts); OWEN, *supra* note 209, at 343, 345–46 (observing that a close reading of comments i, j, and k to section 402A shows they were drafted to "address a single, narrow topic — the responsibility of sellers of products like food, alcoholic beverages, tobacco, and drugs containing inherent product dangers" and that Dean Prosser was "too rushed" to redraft those comments when expanding section 402A to the sale of all products); *cf.* Peter Barton Hutt, *Food and Drug Law: A Strong and Continuing Tradition*, 37 FOOD & DRUG COSMETIC L.J. 123, 132–33 (1982) ("To the extent that there have been problems [with FDA regulation], they have been of scientific rather than of statutory origin. . . . One of the major lessons discernible from the past 75 years is the need to proceed cautiously and deliberately in the face of major scientific uncertainty and controversy.").

218. *See* RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY §§ 7, 20 (A.L.I. 1998); *see also* RESTATEMENT (SECOND) OF TORTS § 402A cmt. f (A.L.I. 1965) (limiting the rule as not applying to "the occasional seller of food" such as "the housewife who, on one occasion, sells to her neighbor a jar of jam or a pound of sugar").

219. *See* RESTATEMENT (SECOND) OF TORTS § 406 (AM. L. INST. 1965) ("A manufacturer who in the course of his business as such directly or through a third person gives or lends to another a chattel made by him, is subject to the same liability as if he had sold the chattel."). *But cf. id.* § 402A cmt. l (asserting that strict products liability should not apply to non-sellers).

220. *See* Benjamin C. Zipursky, *Online Defamation, Legal Concepts, and the Good Samaritan*, 51 VAL. U. L. REV. 1, 31 (2016); President & Dirs. of Georgetown Coll. v. Hughes, 130 F.2d 810, 812–13 (D.C. Cir. 1942) ("Generally also charity is no defense to tort . . . . One who undertakes to aid another must do so with due care.").

221. *See* Carl Zollman, *Damage Liability of Charitable Institutions*, 19 MICH. L. REV. 395, 395–97 (1921) (discussing immunity of municipalities and other governmental agencies); *see also* JANET POPPENDIECK, BREADLINES KNEE DEEP IN WHEAT: FOOD ASSISTANCE IN THE GREAT DEPRESSION 129–39 (1986) (describing food donations from the Federal Farm Board Stabilization Program and the subsequent creation of the Federal Surplus Relief Corporation during the Great Depression). *But see* Jonathan R. Seigel, *Waivers of State Sovereign Immunity and the Ideology of the Eleventh Amendment*, 52 DUKE L.J. 1167, 1187–88 (2003) (arguing that before 1945, state sovereign immunity was more easily deemed to have been waived by implication).

222. *See* Bradley C. Canon & Dean Jaros, *The Impact of Changes in Judicial Doctrine: The Abrogation of Charitable Immunity*, 13 L. & SOC'Y REV. 969, 971 (1979) (explaining that the charitable immunity doctrine was first adopted by Massachusetts in 1876, and then spread to seven states by 1900, twenty-five by 1920, and forty states by 1938); Lester W. Feezer, *The Tort Liability of Charities*, 77 U. PA. L. REV. 191, 197 (1928) (observing that "the

Massachusetts, the defense of charitable immunity spread to about forty states before reversing course and being abrogated by most jurisdictions beginning in the 1940s.[223] The reasons for rejecting charitable immunity were manifold, including recognition that charity is not a license to cause harm with impunity.[224] Other reasons included shifts in tort policy toward victim compensation and the more ready availability of liability insurance.[225]

As common law immunities faded away by mid-century, policymakers began to worry that food donation activities would be deterred by liability threats. State legislatures responded by enacting a patchwork of statutory shields for food donation activities. California led the way in 1977, followed by Oregon and Washington in 1979; within ten years, all fifty states had enacted a good Samaritan food donation law.[226]

Tellingly, the state-by-state approach fell short of resurrecting a total immunity shield. Different states set different eligibility thresholds for immunity. California refused to exempt injuries resulting from "gross negligence or willful act."[227] Oregon set the bar at "gross negligence, recklessness, or intentional conduct."[228] Other states adopted a

---

immunity extends only to institutionalized charity and not to the individual Good Samaritan"). Some states explicitly rejected the doctrine even during its heyday. *See, e.g.*, Geiger v. Simpson Methodist-Episcopal Church, 219 N.W. 463, 465 (Minn. 1928) ("We do not think it would be good public policy to relieve [charitable institutions] from liability for torts or negligence.").

223. *See* DOBBS ET AL., *supra* note 114, at 598 ("Most American courts or legislatures have now rejected [charitable] immunity. The Restatement simply says no such immunity exists."); Canon & Jaros, *supra* note 222, at 972 (noting that "a counter trend began" when the charitable immunity doctrine was rejected in "a devastating opinion" in *Georgetown Coll.*, 130 F.2d 810); *see also* Nicholson v. Good Samaritan Hosp., 199 So. 344, 373–74 (Fla. 1940) (earlier identification of "the modern trend" toward rejection of the charitable immunity doctrine). *But see* Note, *The Quality of Mercy: "Charitable Torts" and Their Continuing Immunity*, 100 HARV. L. REV. 1382, 1385 (1987) (arguing that "[charitable immunity's] decline has been exaggerated").

224. *See* Note, *Quality of Mercy*, *supra* note 223, at 1387, 1391–92 ("[T]he overall goodness of a wrongdoer does not, in law, excuse the wrong."); *Georgetown Coll.*, 130 F.2d at 814 ("It is a strange distinction, between a charitable institution and a charitable individual, relieving the one, holding the other, for like service and like lapse in like circumstances . . . . The basis of the distinction cannot be charity.").

225. *See* Charles Robert Tremper, *Compensation for Harm from Charitable Activity*, 76 CORNELL L. REV. 401, 410–11 (1991) (observing that "[m]uch of the prodding to end charitable immunity came as part of a broader movement to eliminate barriers to tort recovery" and "the advent of inexpensive liability insurance"); Kenneth S. Abraham & Catherine M. Sharkey, *The Glaring Gap in Tort Theory*, 133 YALE L.J. 2165, 2200–02 (2024) (arguing that availability of liability insurance played a key role in abrogation of charitable immunity and of sovereign immunity); *Georgetown Coll.*, 130 F.2d at 827 ("The rule of immunity is out of step with the general trend of legislative and judicial policy in distributing losses incurred by individuals through the operation of an enterprise among all who benefit by it rather than in leaving them wholly to be borne by those who sustain them.").

226. *See* David L. Morenoff, *Lost Food and Liability: The Good Samaritan Food Donation Law Story*, 57 FOOD & DRUG L.J. 107, 109, 112, 116 (2002).

227. *Id.* at 116 (quoting Cal. Food & Agric. Code § 58,505 (West 2000)).

228. *Id.* (quoting Or. Rev. Stat. § 30.890 (1999)).

range of language from ordinary negligence to "actual or constructive knowledge of the harmful quality of the food" or "willful, wanton or reckless act."[229] The upshot is that all states agreed that charitable donors of food should receive special treatment — but none felt there should be impunity from tort liability.

Congress sought to harmonize the state-by-state approach, because it worried that the patchwork of state laws was deterring food donation efforts.[230] An initial effort in 1990 resulted in the passage of a nonbinding Model Good Samaritan Food Donation Act.[231] When the Model Act failed to produce action,[232] Congress took up the issue again and passed the Bill Emerson Good Samaritan Food Donation Act of 1996.[233] In effect, Congress simply replaced "Model" with "Bill Emerson" and converted the model law near-verbatim into binding law.[234]

The Bill Emerson Act states that food donors and nonprofit distributors shall not be subject to liability for donating "apparently wholesome food" in "good faith."[235] The conditional requirement of good faith suggests that this immunity is available only for actors who act honestly — or perhaps reasonably — to comply with food safety laws

---

229. *Id.* (collecting statutory language) (citing WINTHROP, STIMSON, PUTNAM & ROBERTS, SUMMARY OF GOOD SAMARITAN FOOD DONATION STATUTES 11–12, Part D (1992)); H.R. REP. NO. 104-661, at 3 (1996) (observing that some states provide for liability only for "gross negligence or intentional acts," or provide immunity based on reasonable inspection and no "actual or constructive knowledge," while other states "retain liability for negligence and eliminate it only for lawsuits based solely on strict liability").

230. H.R. REP. NO. 104-661, at 3 (1996) ("Private companies are too often faced with different state laws governing food donation. These differences can stand between a willing donor and a needy family. I urge this Subcommittee to lift this barrier so that this assistance can continue and perhaps grow, thereby helping needy families." (quoting Rep. Bill Emerson)); Jessica A. Cohen, *Ten Years of Leftovers with Many Hungry Still Left Over: A Decade of Donations Under the Bill Emerson Good Samaritan Food Donation Act*, 5 SEATTLE J. SOC. JUST. 455, 470 (2006) ("One factor that pushed the [Bill Emerson] Act through Congress, for example, was Wal-Mart's failure to donate due to fear of liability."); Morenoff, *supra* note 226, at 120.

231. Pub. L. No. 101-610 §§ 40–403, 104 Stat. 3127, 3183–85 (1990) (codified at 42 U.S.C. §§ 12671–12673); *see also* Morenoff, *supra* note 226, at 117–18 (observing that the Model Act received support because "it did not require states to enact the Model Act, but only encouraged them to consider it" (citing 136 CONG. REC. S1892 (daily ed. Mar. 1, 1990))). At the same time, Congress turned down efforts to shield a wider range of nonprofit organizations and volunteers. *Id.* at 118–19.

232. 142 CONG. REC. H7480 (daily ed. July 12, 1996) (statement of Rep. Conyers) (observing that only one state had adopted the Model Act).

233. Pub. L. No. 104-210, 110 Stat. 3011 (1996) (codified at 42 U.S.C. § 1791).

234. H.R. REP. NO. 104-661, at 5 (noting that section 1 of the bill converts the model Act to permanent law). There were only two substantive amendments. The first extended immunity to include both food donors and nonprofit distributors. *See id.* at 4–5. The second broadened the definition of "gross negligence" to include failures to act, and to specify that nothing in the bill should be construed to supersede state or local health regulations. *See* Morenoff, *supra* note 226, at 125.

235. 42 U.S.C. § 1791(c). The immunity applies to any person when donating through a nonprofit distributor (such as a food bank or shelter), but is restricted to "qualified direct donor[s]" (e.g., grocers, agricultural producers, or restaurants) when donating directly to needy individuals. *Id.*

and regulations.[236] Separately, the Bill Emerson Act also excludes from immunity any injury resulting from acts or omissions constituting "gross negligence" or "intentional misconduct."[237] The federal government has interpreted the latter provision as a partial preemption for those state laws that set a liability floor below gross negligence.[238] But that interpretation is contestable and has not been tested in court.[239]

Regardless whether federal law sets the threshold at gross negligence, good faith, or some other standard of care, it is clear that Congress did not intend to provide a total exemption from ordinary principles of tort law.[240]

B. Blood Donations

In the medical realm, donations of blood and other biological transplants are generally held to a negligence standard.[241] Such donations carry latent risks of transmitting undetected infections such as hepatitis

---

236. *Compare* Jane Stapleton, *Good Faith in Private Law*, 52 CURRENT LEGAL PROBLEMS 1, 8 (1999) ("[A] standard of reasonable behaviour is more demanding than the requirement of good faith . . . . This critical distinction is confirmed by well-known statutory definitions of good faith which state that 'a thing is deemed to be done in good faith . . . when it is in fact done honestly, whether it is done negligently or not.'"), *with* Jay M. Feinman, *Good Faith and Reasonable Expectations*, 67 ARK. L. REV. 525, 526 (2014) (arguing that "[g]ood faith is simply another embodiment of the basic principle of contract law — the protection of reasonable expectations"), *and* Sarah Munger, Note, *Bill Emerson's Makeover: Reforming the Bill Emerson Good Samaritan Food Donation Act*, 19 VT. J. ENV'T. L. 64, 74 (2018) ("If an objective definition of good faith applies to the Act, the food donor must have donated food that a reasonable person would have thought was 'apparently wholesome' to receive protection."). *But cf.* Rossi v. Motion Picture Ass'n of Am. Inc., 391 F.3d 1000, 1004 (9th Cir. 2004) ("When enacting the DMCA, Congress could have easily incorporated an objective standard of reasonableness. The fact that it did not do so indicates an intent to adhere to the subjective standard traditionally associated with a good faith requirement.").

237. 42 U.S.C. § 1791(c)(4).

238. *See* Morenoff, *supra* note 226, at 128 (citing interpretation of the Department of Justice, which was then adopted by the Department of Agriculture). *But see* Shelden D. Elliott, *Degrees of Negligence*, 6 S. CAL. L. REV. 91, 113 (1933) (observing that the majority of legal theorists reject the concept of degrees of negligence in favor of a simple uniform concept of negligence); RESTATEMENT (THIRD) OF TORTS: LIAB. FOR PHYSICAL AND EMOTIONAL HARM § 2 cmt. a (A.L.I. 2010) (explaining that gross negligence "simply means negligence that is especially bad" and "carries a meaning that is less than recklessness").

239. *See* Morenoff, *supra* note 226, at 128 ("To date, the judicial branch has not had the occasion to provide a definitive answer to this preemption question."); *id.* at 131 (noting that none of the state or federal food donation laws have ever been challenged in court); Cohen, *supra* note 230, at 474 (observing in 2006 that "there have been no documented lawsuits against a food donor either before or after the Good Samaritan Act was passed").

240. *See* 142 CONG. REC. S9532 (daily ed. Aug. 2, 1996) (statement of Sen. Ted Kennedy) ("Any exceptions to the general rules of tort liability must be narrowly tailored . . . . I am satisfied that the standard contained in this bill still requires that food donors and food banks exercise care to ensure that the food they donate or distribute does not harm the people receiving the food.").

241. *See* OWEN, *supra* note 209, at 1050 (noting that almost all American jurisdictions have enacted "blood shield statutes" that prohibit strict liability actions but allow negligence actions).

or HIV/AIDS.[242] A few early court opinions analogized sales of "bad" blood to those of defective food and sought to apply a strict products liability standard.[243] Yet, most other courts disagreed, reasoning that contaminated blood transfusions were more akin to a service than a sale of goods.[244] Likewise, state legislatures reacted by overwhelmingly enacting blood shield laws that either prohibited strict liability actions or explicitly required a finding of fault.[245]

Public policy rationales for rejecting strict liability include the essential need for blood transfusions, deference to "sound medical judgment," and the inability of scientific methods to guarantee that blood is free of dangerous contaminants.[246] At the time, the medical community viewed infection through blood transfusion to be an inevitable but acceptable risk of the life-saving procedure.[247] Accordingly, many courts expressed concerns that holding blood banks liable for undetectable risks would bankrupt an industry serving an essential societal function.[248] Courts interpreting the Restatement also treated blood as a

242. The U.S. has required all donations of whole blood to be unpaid since the 1970s, on the premise that blood from paid donors is less likely to be safe. However, blood banks, hospitals, and other intermediaries can, and do, charge fees for their services. *See* Robert Slonim et al., *The Market for Blood*, 28 J. ECON. PERSPS. 177, 184–85 (2014).

243. *See* Cunningham v. MacNeal Memorial Hosp., 266 N.E.2d 897, 902 (Ill. 1970) (abrogated by statute); *see also* Cmty. Blood Bank, Inc. v. Russell, 196 So.2d 115, 119 (Fla. 1967) (Roberts, J., concurring) (analogizing undetectable viruses in blood to undetectable adulterations in food); Marc A. Franklin, *Tort Liability for Hepatitis: An Analysis and a Proposal*, 24 STAN. L. REV. 439, 474 (1972) (analogizing blood bank to specialized manufacturer and hospital to diversified retailer).

244. *See* Perlmutter v. Beth David Hosp., 123 N.E.2d 792, 795 (N.Y. 1954); *see also* Daniel A. Gioia, Comment, *Blood Transfusions and the Transmission of Serum Hepatitis: The Need for Statutory Reform*, 24 AM. U. L. REV. 367, 392 (1975) ("Contaminated blood transfusions fit neatly into neither category, but rather should be characterized as a unique combination of both a sale and a service."). Software shares this dilemma. *See* Choi, *supra* note 25, at 66–67.

245. *See* RESTATEMENT (THIRD) OF TORTS: PROD. LIAB. § 19 cmt. c (A.L.I. 1998) ("[L]egislation in almost all jurisdictions limits the liability of sellers of human blood and human tissue to the failure to exercise reasonable care . . . . Where legislation has not addressed the problem, courts have concluded that strict liability is inappropriate for harm caused by such product contamination."); Michael J. Miller, Note, *Strict Liability, Negligence, and the Standard of Care for Transfusion-Transmitted Disease*, 36 ARIZ. L. REV. 473, 488–89 (1994) (collecting blood shield statutes in 48 states); George W. Conk, *Is There a Design Defect in the Restatement (Third) of Torts: Products Liability?*, 109 YALE L.J. 1087, 1099 (2000) ("In 1965, three states had blood shield laws; by 1972, the count was forty-one.").

246. *See* McDaniel v. Baptist Mem'l Hosp., 469 F.2d 230, 234–35 (6th Cir. 1972); Heirs of Fruge v. Blood Servs., 506 F.2d 841, 844–45 (5th Cir. 1975); Miller, *supra* note 245, at 490 (collecting legislatures' stated policy reasons); Gioia, *supra* note 244, at 405–06.

247. *See* Conk, *supra* note 245, at 1108–09 (describing "culture of inevitability about illness among hemophiliacs").

248. *See, e.g.*, *Perlmutter*, 123 N.E.2d at 795 (refusing to find liability because "it would mean that the hospital, no matter how careful, no matter that the disease-producing potential in the blood could not possibly be discovered, would be held responsible, virtually as an insurer, if anything were to happen to the patient as a result of 'bad' blood"); *In re* Rhone-Poulenc Rorer, Inc., 51 F.3d 1293, 1298 (7th Cir. 1995) (decertifying mass tort class action); Zichichi v. Middlesex Mem'l Hosp., 528 A.2d 805, 810 (Conn. 1987); Garvey v. St. Elizabeth Hosp., 697 P.2d 248, 250 (Wash. 1985).

"classic example" of an "unavoidably unsafe product[]," removing it from the strict liability paradigm.[249]

Notwithstanding those important public policy concerns, however, the negligence rule has been quite uncontroversial. As with food donations, the most salient issue is not whether negligence is appropriate, but which version of negligence should apply.[250] The dominant view holds that collection and processing of blood is a medical service subject to a professional malpractice standard.[251] If the professional standard applies, then compliance with FDA and industry standards is a complete defense.[252] However, as scientific techniques have improved in recent decades, several courts have ruled that blood banks are expected to exercise ordinary reasonable care, allowing juries to second-guess the reasonableness of standard blood banking practices.[253] Another set of courts has quietly backed into the ordinary care standard, by purporting to apply professional care while asserting that the entire profession's standards can be held unreasonably deficient.[254] The latter

---

249. *See* Rogers v. Miles Lab'ys, Inc., 802 P.2d 1346, 1351 (Wash. 1991) ("Justice Traynor clearly saw comment *k* as applying to blood and blood derivatives." (quoting Roger J. Traynor, *The Ways and Meanings of Defective Products and Strict Liability*, 32 TENN. L. REV. 363, 367–68 (1965))); RESTATEMENT (THIRD) OF TORTS: PROD. LIAB. § 19(c) (A.L.I. 1998) (excluding human blood and human tissue); *see also* Miller, *supra* note 245, at 486 (collecting cases).

250. *See* Patricia Kussmann, Annotation, *Validity, Construction, and Application of Blood Shield Statutes*, 75 A.L.R.5th 229 § 2[a] (2000); Kathryn W. Pieplow, Comment, *AIDS, Blood Banks and the Courts: The Legal Response to Transfusion-Acquired Disease*, 38 S.D. L. REV. 609, 620–21 (1993).

251. *See* Kussmann, *supra* note 250, §§ 14, 17[a] (collecting cases); Pieplow, *supra* note 250, at 626–29; Miller, *supra* note 245, at 509–10 (collecting cases).

252. *See* Giorno v. Temple Univ. Hosp., 875 F. Supp. 267, 269 (E.D. Pa. 1995) (asserting that the professional care standard means plaintiff must prove defendant's failure to comply with established blood banking safety standards); *see also* Choi, *supra* note 28, at 597 (arguing that "the 'professional' label . . . shift[s] the standard of care from reasonable care to customary care," which "restricts the jury inquiry to assessing compliance with the profession's internal custom").

253. *See* Miller, *supra* note 245, at 510 (some courts subject blood banks to an ordinary standard of care because "absolute adherence to a professional standard of care would allow blood banks to determine their own legal duty, which is the proper function of the courts"); Doe v. Cutter Biological, Inc., 971 F.2d 375, 382–83 (9th Cir. 1992) (noting that custom, usage, or industry practice may fall short of reasonable care, especially when the entire industry is comprised of a handful of manufacturers); Doe v. Am. Nat'l Red Cross, 798 F. Supp. 301, 306 (E.D.N.C. 1992) (statutory language of "due care" unequivocally makes blood banks "subject to an ordinary standard of negligence, rather than a professional standard of negligence"); Ray v. Am. Nat'l Red Cross, 696 A.2d 399, 404–05 (D.C. 1997).

254. *See* Miller, *supra* note 245, at 510–11 ("These courts have held that the professional standard does apply to blood banks, but that it can be refuted by evidence that the whole industry failed to adopt the best procedure. This position is inconsistent."); Doe v. Am. Nat'l Red Cross, 848 F. Supp. 1228, 1233 (S.D. W. Va. 1994) (purporting to apply a professional negligence standard, but holding that "[c]ustomary practice does not prescribe the duty of care"); United Blood Servs. v. Quintana, 827 P.2d 509, 522 (Colo. 1992) (en banc) (finding that Colorado's blood shield statute "imposes a professional standard of care on a blood bank" but that "under the particular circumstances of this case, such standard should not have been

position is likely motivated by pointed evidence that the blood transfusion industry deliberately ignored readily available safeguards.[255]

Another distinction from the food donation cases is that the defendants in blood donation cases are primarily intermediary distributors such as blood banks or hospitals, and not the direct donors.[256] Nevertheless, when the question of identifying individual donors has been raised, courts have been split.[257] Courts that deny discovery of donor identities have worried that undue identification could deter the "free flow" of donor activity[258] or violate donors' privacy rights.[259] But more recent decisions have evinced greater willingness to require disclosures as needed.[260]

V. THE THREE MODULES OF OPEN-SOURCE LIABILITY

The law of tainted donations offers three broad lessons for correcting the reflexive immunity conferred upon open-source software.

---

considered as conclusive proof of due care"); Advincula v. United Blood Servs., 678 N.E.2d 1009, 1027–28 (Ill. 1996) (holding that "conformance with professional standards of care, proven by expert testimony or other evidence of professional standards, is indicative but not conclusive of due care"); Kirkendall v. Harbor Ins. Co., 887 F.2d 857, 860–61 (8th Cir. 1989) (finding that contemporaneous industry practice is not dispositive for determining applicable standard of care).

255. *See* Linda M. Dorney, Comment, *Culpable Conduct with Impunity: The Blood Industry and the FDA's Responsibility for the Spread of AIDS Through Blood Products*, 3 J. PHARMACY & L. 129, 142–49 (1994) (criticizing persistent refusals by blood banking leaders to implement screening and testing policies); Kieran Healy, *The Emergence of HIV in the U.S. Blood Supply: Organizations, Obligations, and the Management of Uncertainty*, 28 THEORY & SOC'Y 529, 548 (1999) (explaining why blood banks "chose to play down the problem" of transfusion AIDS); *see also* Slonim, *supra* note 242, at 183–84 (noting that "a low-cost procedure to eradicate hepatitis B in plasma developed in the late 1960s was not commonly used until the early 1980s"); Conk, *supra* note 245, at 1109–11 (arguing that blood shield laws slowed development and adoption of safer pasteurization techniques).

256. *See* Franklin, *supra* note 243, at 446–47 ("Few plaintiffs attempt to fasten liability on the donors in hepatitis cases. . . . [E]ven if liability can be established, there is small likelihood that a judgment of damages against an individual can be collected.").

257. *See* Joseph Kelly, *The Liability of Blood Banks and Manufacturers of Clotting Products to Recipients of HIV-Infected Blood*, 27 JOHN MARSHALL L. REV. 465, 470–72 (1994); Note, *Transfusion-Related AIDS Litigation: Permitting Limited Discovery from Blood Donors in Single Donor Cases*, 76 CORNELL L. REV. 927, 929 (1991); *see also* Volokh, *supra* note 190, at 1367, 1405 (observing that fully naming the parties to a suit is the default, and categorizing the situations where that presumption can be rebutted, including "privacy as to 'sensitive and highly personal' 'stigmatized' matters").

258. *See* Krygier v. Airweld, Inc., 520 N.Y.S.2d 475, 476 (N.Y. Sup. Ct. 1987); Ellison v. Am. Nat'l Red Cross, 151 F.R.D. 8, 11 (D.N.H. 1993).

259. *See* Rasmussen v. S. Fla. Blood Serv., Inc., 500 So. 2d 533, 538 (Fla. 1987) (refusing to permit a "fishing expedition"); Doe v. Univ. of Cincinnati, 538 N.E.2d 419, 424 (Ohio Ct. App. 1988); *Krygier*, 520 N.Y.S.2d at 477.

260. *See, e.g.*, Roth v. N.Y. Blood Ctr., Inc., 596 N.Y.S.2d 639, 642 (N.Y. Sup. Ct. 1993); Doe v. Puget Sound Blood Ctr., 819 P.2d 370, 376–77 (Wash. 1991); Watson v. Lowcountry Red Cross, 974 F.2d 482, 488–89 (4th Cir. 1992); Gulf Coast Reg'l Blood Ctr. v. Houston, 745 S.W.2d 557, 560 (Tex. App. 1988).

The starting proposition is that direct donors of code are not entitled to blanket immunity based on eleemosynary principles of volunteerism or social benefit. Instead, even the most benevolently intentioned actor typically remains subject to ordinary negligence principles. While many open-source contributors may be "judgment-proof" because they lack assets or cannot be hailed in court, it is nonetheless consequential to ground liability in the direct donative act.[261] A light-touch approach made sense when the open-source community was nascent and the primary challenge was growth. But the modern open-source ecology indicates that the hydraulics have shifted: the critical problem today is an excess volume of low-quality projects and contributions. A rising tide of liability risk could help lift coding standards by discouraging casual amateurism and professionalizing the dedicated maintainers who do the bulk of the work.[262]

The second, more potent proposition is that code-hosting platforms are subject to supply chain liability. Sites that host open-source code are intermediary donees of code — analogous to food banks or blood banks. A strict theory of liability would impute primary liability directly onto the intermediary. But even a negligence-based theory can be founded on independent duties of care such as monitoring, detection, and removal of harmful materials.[263] Examples of minimum safeguards could include identity verification, code reviews, vulnerability scans, and SBOM generation. Honing liability rules for code-hosting platforms would likely be more efficient than pursuing scattered claims against individual open-source contributors, because the platform's centralized role provides economies of scale and makes it the cheapest cost avoider for certain oversight duties.

Third, the analogy between donations of code and donations of food and blood begins to break down where the former serves more purely informational uses and the latter serve more functional needs. This disjunction indicates the need to define a safe harbor that exempts bona fide expressive sharing of code.

A narrow form of the safe harbor could be limited to academic discourse. Drawing this line based on authorship is difficult because there is fluid intermixing in computer science between university and

---

261. *Cf.* Limelight Networks, Inc. v. Akamai Techs., Inc., 572 U.S. 915, 921 (2014) (ruling under the Patent Act that secondary "liability for inducement must be predicated on direct infringement").

262. The history of ham radio operators offers an instructive lesson on the shift from amateurism to professionalization. *See generally* KRISTEN HARING, HAM RADIO'S TECHNICAL CULTURE 89, 96 (2007) (describing the transition in ham radio culture from one that celebrated amateurism to one that became an occupational profession).

263. For example, food banks and blood banks are expected to inspect for contamination the donations they receive. *See also* Tiffany (NJ) Inc. v. eBay Inc., 600 F.3d 93, 106–09 (2d Cir. 2010) (dismissing secondary trademark infringement claims based on defendant's reasonable efforts to detect and remove contraband goods).

industry research. Instead, adding friction to open-source sharing could be a more effective way to distinguish the purpose of use.[264] For example, academic code could be placed on sequestered platforms clearly designated for research purposes only and not for commercial deployment. Because commercial platforms like GitHub are designed to make code reuse simple and easy, placing an "air gap" between code-hosting platforms would prevent accidental commingling of academic code with deployed code, while also avoiding claims of insufficient warning or notice that the academic code is unsuited for commercial use. To be sure, adding friction in these ways would not necessarily prevent copying or transfer of academic code to commercially deployed systems. Where code safety is more critical, access could be restricted only to credentialed researchers, similar to many academic journals.

A broader form of speech exception could borrow from the professional speech framework that applies to learned professions such as medicine and law.[265] There, too, speech is commingled with conduct, such that the professional's speech interests are greater when conversing with peers for purposes of knowledge-generation, and diminished when giving advice to clients who are passive recipients of the professional's expertise.[266] Applying the professional speech rule to open-source software would extend the safe harbor from academic researchers to anyone qualifying as a software professional.[267] Once again, the distribution of executable code would necessarily be excluded from the scope of this protection.

These three components — negligence for code donors, negligence for code banks, and safe harbors for research communities — constitute essential building blocks for establishing a law-based foundation of open-source accountability.

A. Code Donors

If the law of tainted donations makes clear that some form of negligence should apply, it still leaves murky which version is most appropriate for software. Food donors are held to either a gross negligence standard or a good faith standard, depending on one's statutory interpretation. Blood donors are subject to diverse state-by-state laws that

---

264. *See supra* note 140.

265. *See* Claudia E. Haupt, *Professional Speech*, 125 YALE L.J. 1238, 1238 (2016); *see also* Choi, *supra* note 28, at 589.

266. *See* Haupt, *supra* note 265, at 1243–44.

267. A software professional would not necessarily need to be formally accredited but should be defined instead by the nature of the work being performed. *See generally* Choi, *supra* note 28, at 600, 614. *But see* Chinmayi Sharma, *AI's Hippocratic Oath*, 102 WASH. U. L. REV. 1101, 1159–64 (2025) (offering the conventional view that the path to professionalization depends on a formal licensing regime, in addition to higher education programs, professional associations, and codes of conduct).

range somewhere between a professional negligence standard and an ordinary negligence standard. And as a general matter, charitable actors are subject to ordinary negligence rules. These doctrinal choices are situated in the greater context that donations of food, blood, and charity are all highly valued virtues of modern society.

The general prevailing norm is that donors are held to the same standard as non-donors. Good Samaritans who volunteer assistance are treated no differently than any actor who owes a general duty of care. Charitable entities no longer receive special immunities from liability. Likewise, where blood donations are held to a professional negligence standard, the principal justification is that blood bank workers are a type of healthcare professional and should be held to the same medical malpractice standard as other healthcare professionals. Thus, aligning software liability law with the broader trend in tort law would favor either an ordinary negligence or professional negligence standard, depending on how software work is classified.

Food law presents an important contrast to the usual rule, setting a different bar for food donors versus ordinary food manufacturers and vendors. It is certainly plausible to attribute the different outcome to political economy and legislative lobbying by large food vendors. But a more principled distinction could also point to the fact that the food supply chain generates oversupply and moral outrage over wastage of perfectly good food. Even with the lenient liability standard, excess food tends to be destroyed rather than donated. Another important distinction is the presence of well-developed food safety codes that make it easier to understand when food is unfit for consumption, and to have high trust in the food supply chain. By contrast, the blood supply chain grapples with the opposite problem of undersupply and moral panic about receiving low-quality contributions. And although the science of blood safety and blood testing has improved substantially, it still carries an element of unknowable risk.

Thus, the choice of rule for open-source liability could depend on whether policymakers believe the principal problem is one of oversupply or of undersupply. Most old-school open-source advocates believe there is no such thing as too much free code. But a newer, contrary movement points out that a vast proportion of free contributions are amateurish and unfit to be committed to production quality code. The growing mood among observers is that there is simultaneously an oversupply of low-quality code, and an undersupply of skilled maintainers of code. Nor are there effective software safety codes that could make it easier to spot bad code. Despite a plethora of so-called standards, risk management frameworks, and certificate programs, the practice of

software development remains deeply fragmented with strong ideological disagreements that lack evidence-based paths to resolution.[268]

The closer analogous fit for software supply chains is likely to be to blood donation supply chains. In addition to the undersupply problem, code donations resemble blood donations in that much of the legal scrutiny is on intermediary donees — the project maintainers and blood handlers — and not on the primary donors. Some observers have argued that a reasonableness standard can be readily extended to software liability determinations.[269] Yet, there are also good reasons to believe the professional negligence standard would be the better fit.[270] Those arguments have been rehearsed elsewhere and rest on the complexity and heterogeneity of software, as well as the absence of objective consensus among the expert community regarding good software development practices.[271]

Alternatively, it can be argued that the food donation supply chain offers the better paradigm. Unlike blood donations, neither food donations nor open source has attracted significant litigation. And policymakers have long favored the narrative that the distribution of free code needs to be further incentivized, not metered. If that bifurcated model is preferred, then the gross negligence test appears more viable than a good faith compliance test, because software safety is a less mature regulatory environment than food safety. Good faith compliance depends on the existence of mature regulatory codes that provide high confidence. In contrast, gross negligence merely sets a minimum floor of avoiding obvious worst practices, for which there is already emerging guidance.

B. Code Banks

Holding code-hosting platform operators — or "code banks" — to a negligence standard opens a different range of policy interventions than bringing suit against individual project maintainers.[272] The use of

---

268. *See generally* Choi, *supra* note 31; Choi, *supra* note 28, at 567.

269. *See* Sharma & Zipursky, *supra* note 28 (touting the products liability framework, which "utilizes the notion of 'reasonableness'"); *see also supra* note 28 and accompanying text.

270. *See* Choi, *supra* note 28, at 603–05 (explaining that professional negligence requires juries to defer to expert evidence of professional custom, whereas ordinary negligence allows juries to impose their own independent judgment of reasonableness); Sharma, *supra* note 267, at 1120–26; *see also* Bambauer & Teplinsky, *supra* note 30 (advocating for more concrete guidance on standard of care for software developers because just "reasonableness" could be "uncertain and unpredictable").

271. *See* Choi, *supra* note 28, at 576, 602, 625.

272. *See* Rory Van Loo, *The Revival of Respondeat Superior and Evolution of Gatekeeper Liability*, 109 GEO. L.J. 141, 161 (2020) ("[P]latforms' inherent ease of monitoring and blocking users can be persuasive evidence of agency."); *cf.* Paul Ohm, *Regulating at Scale*, 2 GEO.

intermediary powers invokes longstanding debates regarding collateral censorship and the proper role of intermediary liability.[273] To be sure, there are some interactive aspects of code banks that resemble other prominent social media platforms. The duties of care contemplated here are limited to functional safeguards and factual labels, and do not extend to editorial mandates.[274]

While code banks could be held secondarily liable for the coding errors of their users, a more nuanced approach would construct independent duties of oversight and supervision that are better tailored to the capabilities and overheads of platform entities. In other analogous areas, intermediary donees such as food banks and blood banks are subject by statute to the same liability standard as primary donors. But this crude lumping could be parsed apart. Lawmakers need not apply the same standard of care to donor individuals and to donee platforms, particularly where the latter are ordinary commercial entities charging fees and seeking to maximize profit.[275] For example, GitHub was acquired by Microsoft in 2018 and has been criticized for failing to uphold open-source values.[276]

Like food banks and blood banks, code banks can use frictional measures to block potentially bad contributions. A basic place to start is identity verification.[277] While many swaths of the internet permit

---

L. TECH. REV. 546, 553 (2018) ("Most fundamentally, some standards of care ought to be subjected to an 'orders of magnitude' rule. For every order of magnitude growth in the number of users, the standard of care to which you are subjected should grow at a faster-than-linear rate.").

273. *See supra* Part III.C; *see also* Wu, *supra* note 75, at 295–97; Jack M. Balkin, *Free Speech Is a Triangle*, 118 COLUM. L. REV. 2011, 2017–19 (2018).

274. *See* Moody v. NetChoice, LLC, 603 U.S. 707, 727–28 (2024) (stating that the First Amendment offers protection against "restrictions on the platforms' selection, ordering, and labeling of third-party content" if and only if "the regulated party is engaged in its own expressive activity, which the mandated access would alter or disrupt"); *id.* (Barrett, J., concurring) ("A function qualifies for First Amendment protection only if it is inherently expressive."); R.J. Reynolds Tobacco Co. v. FDA, 96 F.4th 863, 877 (5th Cir. 2024) (holding that the First Amendment does not prohibit "purely factual and uncontroversial" disclosure requirements).

275. *See* Pegram v. Herdrich, 530 U.S. 211, 234–35 (2000) (holding health maintenance organization decisions to a different standard than decisions made by medical professionals); *cf.* Gad Weiss, *Aligned Structuring of AI Startups*, 58 ARIZ. ST. L.J. (forthcoming 2026) (explaining that even attempts to structure corporations in "alignment" with societal values such as safety are not reliably effective).

276. *See* EGHBAL, *supra* note 19; *see also GNU Ethical Repository Criteria Evaluations*, FREE SOFTWARE FOUND. (Apr. 25, 2022, at 15:33 ET), https://www.gnu.org/software/repo-criteria-evaluation.html [https://perma.cc/V5K5-HTUX] (quoting Tom Ryder, *Why Not GitHub?*, SANCTUM.GEEK, https://sanctum.geek.nz/why-not-github.html [https://perma.cc/Y9JY-8QEP] (Apr. 26, 2023, at 05:11 UTC)); *Give Up GitHub!*, SOFTWARE FREEDOM CONSERVANCY, https://sfconservancy.org/GiveUpGitHub/ [https://perma.cc/C9XY-CSMD].

277. *Cf.* 20 C.F.R. § 401.200 (authorizing States to "require a blood donor to furnish his or her social security number when donating blood"); *Acceptable Forms of ID for Blood Donors*, AM. RED CROSS, https://www.redcrossblood.org/donate-blood/blood-donation-process/be

anonymous engagement — and are better for it[278] — software development is a special case.[279] Because the functional aspects of code so often dominate the expressive aspects, most uses of code banks are more akin to fixing up a car than to distributing pamphlets about vehicle modifications. Recharacterizing open-source participation as a conditional privilege like driving, rather than as a right like speech, would likely reduce the prevalence of bad and malicious code contributions. To be clear, identification is not equivalent to a licensure scheme; individuals would remain free to self-publish code or to participate on less reputable (or extra-jurisdictional) platforms. But deplatforming from major publication venues would reduce the circulation and reach of shoddy code.[280]

Other reasonable duties for large code banks might include low-level safeguards such as requiring redundant staffing of maintainers for "critical" projects, or in the alternative flagging when projects are at risk of becoming inactive or abandoned. Studies show that a supermajority (more than sixty percent) of projects are maintained by just one or two developers, and that about eighteen percent of projects stop being maintained in a given year.[281] It would probably be too burdensome to require redundant staffing for every new open-source project, but it may be reasonable to require it soon after a project reaches some level of criticality.[282]

Such assessments of project health could be supplemented by automated tools such as vulnerability scanning, dependency management, and SBOM generation. Such tools are still in early development, however, and their feasibility may also depend on the code bank's size and resources.

C. Speech Harbors

Although the foregoing discussion rejects a general immunity approach for open source, there are several reasons to consider a limited-purpose safe harbor for academic or scientific discourse.

The first is legal doctrine, which has rightly held that the government may not restrict disclosures of source code that constitute

---

fore-during-after/acceptable-formsofidforblooddonors.html [https://perma.cc/D83N-7XRE] (requiring valid and unexpired forms of ID).

278. *See* Cohen, *supra* note 189, at 981–82.

279. *See* Sharma, *supra* note 270, at 1125–27, 1162–64.

280. *See* Ganesh Sitaraman, *Deplatforming*, 133 YALE L.J. 497, 542–44, 556 (2023).

281. *See* EGHBAL, *supra* note 144, at 52; SONATYPE 2023 REPORT, *supra* note 1, at 21; *see also* EGHBAL, *supra* note 19, at 108 (describing one maintainer's practice of transferring control of projects to a "ghost" admin account once "he gets tired of maintaining them" so that there is "literally nobody at the helm").

282. *See* EGHBAL, *supra* note 19, at 65 (observing that because "maintenance often requires knowledge that isn't easily externalized to others," the longer maintainers "go without externalizing this knowledge, the more difficult it becomes for newcomers to participate").

communication of expressive ideas.[283] To be clear, not every publication of source code qualifies as an expressive act; otherwise, the exception would swallow the rule. Rather, the dissemination must be in support of a bona fide presentation of academic work to a research community. Distributing code in a form that can be readily executed outside that context would escape that narrow exception.[284]

A second, more pragmatic justification is that scientific norms are shifting in the direction of expecting and incentivizing open publication of code and data in order to facilitate reproducibility of research. Major academic conferences and journals have begun to encourage or require disclosure as part of their submission review processes.[285] Likewise, grant-making institutions such as the National Science Foundation encourage principal investigators to deposit software and data created under federally funded awards.[286] If the primary purpose of distributing academic software is to promote research integrity, scientific reproducibility, and peer review — rather than to create real-world systems — then the threat of liability for releasing imperfect code is counterproductive to those purposes.

---

283. *See supra* Part III; *see also Bernstein*, 176 F.3d at 1145 (9th Cir. 1999) ("To the extent the government's efforts are aimed at interdicting the flow of scientific *ideas* (whether expressed in source code or otherwise), as distinguished from encryption *products*, these efforts would appear to strike deep into the heartland of the First Amendment.").

284. *See* Samuelson, *supra* note 162.

285. *See* Lorena Barba, *The Path to Frictionless Reproducibility Is Still Under Construction*, 6 HARV. DATA SCI. REV. 1, 2–3 (2024), https://hdsr.mitpress.mit.edu/pub/7ncz09ji/release/1 [https://perma.cc/M84Y-TUR4] (explaining that, circa 2019, major computer science conferences began encouraging open code and data submissions); *ICML 2024 Author Instructions*, INT'L CONF. ON MACHINE LEARNING, https://icml.cc/Conferences/2024/AuthorInstructions [https://perma.cc/SM7K-MF4X] ("Authors are encouraged to submit code to foster reproducibility. Reproducibility of results and easy availability of code will be taken into account in the decision-making process."); *NeurIPS Code and Data Submission Guidelines*, NEURAL INFO. PROCESSING SYS., https://neurips.cc/public/guides/CodeSubmissionPolicy [https://perma.cc/SW48-HAGF] ("If any of the main contributions of your paper depends on an experimental result, you are strongly encouraged to submit code that produces this result. If you are using a new dataset, you are also encouraged to submit the dataset."); *Submission Guidelines*, PROCS. VERY LARGE DATA BASES, http://vldb.org/pvldb/volumes/18/submission [https://perma.cc/XCK2-DPPT] ("Authors are expected to submit supplemental material, such as code, data and other implementation artifacts used to produce the results reported in the paper.").

286. *See Proposal and Award Policies and Procedures Guide*, NAT'L SCI. FOUND., https://new.nsf.gov/policies/pappg/24-1/ch-11-other-post-award-requirements#ch11D4 [https://perma.cc/58JD-QJRQ] ("Investigators and recipients are encouraged to share software and inventions created under the award or otherwise make them or their products widely available and usable."); *see also Writing a Data Management & Sharing Plan*, NAT'L INSTS. HEALTH, https://sharing.nih.gov/data-management-and-sharing-policy/planning-and-budgeting-for-data-management-and-sharing/writing-a-data-management-and-sharing-plan#elements-to-include-in-a-data-management-and-sharing-plan [https://perma.cc/Q5PQ-62SC] ("Indicate whether specialized tools are needed to access or manipulate shared scientific data to support replication or reuse, and name(s) of the needed tool(s) and software. If applicable, specify how needed tools can be accessed.").

Relatedly, a third tentative reason to exempt academic software is that much of it is truly "academic" in the sense of being inoperable.[287] Often, such software is written as a proof-of-concept and is not intended or optimized for real-world performance. For open-source code that is clearly intended only for academic discussion, the expressive characteristics override the functional uses. That said, this argument is weakened where the code finds substantial success with a broader audience. As an example: the TinyImages database began as an academic project at MIT but soon became an essential resource for training image recognition systems. When the authors discovered troubling flaws in the database, they opted to take down the entire database rather than allow it to remain available for public use.[288] Likewise, if an academic researcher releases a software package that is then imported directly into a system in widespread use, it may become appropriate at some juncture to reclassify the package as nonacademic work.

Administering such a safe harbor could prove especially challenging if academic code and nonacademic code are intermingled together on the same code-hosting platform. Because such platforms encourage code sharing and reuse through social engagement features, open-source developers of nonacademic projects would be more likely to discover and then import academic code posted within the same platform, regardless of disclaimers or warnings that such use is disallowed.

A more prudent scheme would condition the safe harbor on using a code-hosting platform that is dedicated solely to academic code. For example, academic venues that require submissions to include code and data could offer their own dedicated servers and refuse to accept submissions through GitHub repositories. Alternatively, GitHub or a third-party provider could establish a partitioned service for only academic works. By sharing their work within these limited-purpose venues, a researcher could signal more easily that their work should not be directly incorporated into commercial or real-world systems, and also that their work is eligible for safe harbor from liability.

VI. CONCLUSION

Any serious discussion of software liability must eventually confront the difficult (and unpopular) challenge of open-source liability. Open-source code is ubiquitous in modern software systems, and so

287. *See, e.g.*, David Chanin, *Academics: You're Doing Open Source Wrong*, GITHUB (June 4, 2023), https://chanind.github.io/2023/06/04/academics-open-source-research-code-python-tips.html [https://perma.cc/HG2J-RAW8] ("Most code I encounter that's released as part of academic papers is completely broken, as in it's not possible to run the code as provided at all.").

288. *See Letter from Antonio Torralba, Rob Fergus & Bill Freeman*, TINYIMAGES (June 29, 2020), https://groups.csail.mit.edu/vision/TinyImages/ [https://perma.cc/G926-E73F].

open-source software supply chains have become essential to software development processes. Moreover, because software components are uniquely interdependent and iteratively updated in ad hoc manner, they are unlike physically manufactured components that can be inspected once and tangibly handed off down the supply chain. Thus, there needs to be closer examination of all work processes throughout the entire software supply chain, including open-source maintainers, not just so-called "final" assemblers and vendors.

A study of analogous doctrines in the food and blood donation contexts helps simplify the issue. Like donations of open-source code, donations of food and blood share the twin characteristics of providing high societal value while also being charitable or nonprofit in nature. These overwhelmingly positive characteristics led lawmakers to lobby for shield laws, which aimed to eliminate strict products liability for the donors and donees who sustain these vital supply chains. Yet, the surprising insight is that, in both contexts, lawmakers eschewed a total immunity approach and chose instead to preserve a negligence-based liability framework.

Drawing on those lessons, this Article has proposed a three-part approach to open-source liability: (1) a negligence-based standard of care for open-source maintainers and contributors, (2) complementary duties of supervision for "code bank" platforms and other supply-chain intermediaries, and (3) a limited safe harbor for academic or scientific research. This approach builds on prior work advocating recognition of the same — if not stricter — obligations for proprietary software developers.[289] Ultimately, all software development and maintenance work should be handled with appropriate care. The proposed framework here seeks to close a loophole through which critical segments of the software supply chain would otherwise escape legal oversight.

289. *See generally* Choi, *supra* note 28 (arguing that tort liability for software developers should be based on customary practices of the software community rather than those of an ordinary reasonable person).