WHAT COMES AFTER FOSS?

INKLINGS FROM THREE COMMUNITIES OF PRACTICE

JOHN D. BOY
Leiden University
j.d.boy@fsw.leidenuniv.nl

SUMMARY

Technologists jointly producing and freely sharing software have had a decisive impact on our digital infrastructure. Their work has been shaped not just by changes in technology, but by **changing cultural repertoires** as well. The research underlying this report was motivated by such a cultural shift among technologists: the "**post-FOSS moment**." There are multiple reasons why technologists no longer take the free and open source software movement for granted as the main point of reference for their work. This report proposes a conceptual tripod to make sense of this discontinuity. First, as Big Tech has attained ever greater power over the labor process of developers by using FOSS to set standards, a process of **recursive deskilling** has taken place resulting in bloat, burnout, and a loss of craft. Second, repeated failures of FOSS norms and organizations to resolve problems have brought about a **legitimation crisis** calling the implicit liberalism of FOSS into question. This loss of legitimacy pertains to deliberation as a means of decision-making and to licensing (both copyleft and permissive) as a means of empowerment and control. Third, ingrained habits and norms among technologists have turned into **learning blockages** that manifest in stagnation, regressive behaviors, and backlash. Next, the report presents insights from a workshop with members of **three communities of practice** in which technologists respond to the post-FOSS moment and its learning needs. They work on **permacomputing**, **server collectives**, and **decentralized networking technologies**. A discussion with members of these communities of practice brought to light multiple themes: they agree on the need for a **substantive vision of computing**, they draw inspiration widely from diverse **forms of organization and governance**, they recognize that **vibes matter**, and they seek to foster **intrinsic joy**. Taken together, these themes provide inklings of the collective learning process technologists are engaged in and of the repertoires they will draw on going forward. Recognizing that a cultural shift away from old orthodoxies is happening is a necessary starting point for expanding the map of technical practices that will shape the digital infrastructure of tomorrow.

TABLE OF CONTENTS

Introduction 2
Understanding the "Post-FOSS" Moment 3
  Recursive Deskilling 4
  Legitimation Crisis 5
  Learning Blockages 7
Three Communities of Practice 10
  Permacomputing 11
  Server Collectives 11
  Decentralized Networking Technologies 12
Code as Critique 13
  Taking Back Computing 13
  Growing Together 17
  The Importance of Vibes . . . 19
  Building a Neighborhood . . . 21
Discussion 24
  Key Insights 24
  Further Research 25
  Conclusion 26
Acknowledgements 27
References 27

D// F This research was funded by the **Digital Infrastructure Insights Fund**'s third funding cohort.

July 28, 2026

DOI: 10.17613/5w713-y4n33 · License: CC BY-ND 4.0 International

Introduction

We can now look back on a long history of technologists working together to create software and freely sharing the fruits of their labor with the wider world. Providing foundations for everything from operating systems to digital networks, these technologists decisively shaped today's digital infrastructure. Over the course of this history, how software is made, distributed, and deployed has changed. These changes were driven not only by transformations of technology, but by changing cultural repertoires as well. New organizations and artifacts came on the scene, and new values and terminology caught on. A widely debated change was the shift from "free software" to "open source" as a label for what these technologists were creating. Open source as a term was meant to be more palatable to corporations, and the more permissive licenses championed by the Open Source Initiative were meant to be less off-putting than the copyleft licenses pioneered by the Free Software Foundation.

Scholars who want to understand possibilities for "reordering the technologies and infrastructures that have become part of the fabric of everyday life" have remained interested in these technologists. They documented how many of these technologists, once carriers of utopian, subversive, even radical approaches to using and shaping digital technologies, were routinized and incorporated into industry, but also how some of them continue to embody technopolitical alternatives.

→ Coleman (2011, 515)

This tension does not just mark the scholarship on these technologists—variously called hackers, geeks, or nerds—but their self-reflections as well. The sociologist Ann Swidler observed that "[b]ursts of ideological activism occur in periods when competing ways of organizing action are developing or contending for dominance. People formulate, flesh out, and put into practice new habits of action." Over the past decade, it has been possible to discern signs of such "ideological activism" in the widely expressed sentiment among free and open source software (FOSS) movement insiders that something is amiss. Previously taken-for-granted practices and norms pertaining to FOSS were in question, and there was no clear consensus among movement stakeholders about a way forward. This was already the case before (just to name a recent event from these fast-moving times) a coding agent was used to clone a popular FOSS project and strip it of licensing restrictions. The vibe was that FOSS had "won," but that its victory had come at a steep cost. Licenses and practices of software development and distribution pioneered by FOSS had become dominant in industry, and FOSS projects powered some of the most powerful companies on the planet, but the movement that had started taking shape in fits and starts since the 1980s was a shambles. In the initial proposal for this project, I referred to this as the "pyrrhic victory" of FOSS. The story may appear different now than it did as recently as six months ago, but the underlying sense of crisis and that we are in a moment in which FOSS has been called into question in some fundamental way remains.

→ Swidler (1986, 279)

A small selection of people expressing a version of this sentiment includes Villa (2016); Hoye (2018); Klabnik (2019); Mansoux and Roscam Abbing (2020); Horn (2020); Cotton (2021); Olson (2022); Geuter (2025).

Schrape (2019) describes the journey of FOSS from "niche phenomenon" to "integral part of the software industry."

This report proposes a way to make sense of this moment on the basis of a conceptual tripod. Its three legs are the concepts of recursive deskilling, legitimation crisis, and learning blockages. Each of these concepts helps identify learning needs posed by the present moment

for those who want to further the work of building digital infrastructures that serve the public interest, expand user freedom, and resist capture by regressive forces. It then presents three communities of practice in which these learning needs are taken up to understand how technologists are making a road out of crisis by walking it. These communities of practice work on permacomputing, server collectives, and decentralized networking technologies. This project involved ethnographic research that informed the case selection and conceptual framing. (See Box 1.) It culminated in a workshop involving practitioner-researchers from each of these three communities.

In the workshop, we deepened our understanding of how these communities learn through a series of discussions and activities. Their learning processes are situated and revolve around specific practices, but there also emerged an element of collective learning, of transforming technical practice, that speaks to the wider issues of the "post-FOSS" moment. On the basis of the results of the workshop, I can sketch some tentative responses to the titular question of this report: what comes after FOSS? This is an urgent question for all movement stakeholders, particularly those seeking to build everyday technologies for conviviality and consent rather than extraction and coercion.

Understanding the "Post-FOSS" Moment

As mentioned at the outset, the reasons for the sense of crisis in FOSS are multifarious and evolving. There is no single explanation why movement insiders are reassessing longstanding practices and norms. Any attempt to understand this moment has to be multidimensional. In this section, I introduce three concepts that can help us understand the roots of the sense of crisis—as well as what may be needed to

Box 1: A Note on Methods

The mode of inquiry in this project was ethnographic. Ethnography refers to the systematic study of cultural phenomena from the point of view of those who make and live the cultures being studied. It involves application of a variety of qualitative techniques in the course of longer-term research. Aside from the methodological toolkit of anthropology and sociology, I drew methodological inspiration from foundational studies that scholars in these fields have done on hackers and other communities of technologists, including the ones discussed in this report.

The format of the workshop was inspired by participatory approaches in activity theory. It sought to actively involve practitioners in generating insights into practices they perform while stimulating cross-pollination between different groups to make a positive contribution to collective learning. I derived the main themes of the workshop discussions through a process of qualitative coding of the notes I and other participants took.

In preparation for the workshop, I attended numerous events (e.g., permacomputing meetups, FOSDEM/OFFDEM, MozFest) and had numerous discussions with technologists. I conducted no formal interviews, and the dispersed nature of my interlocutors' work meant that there were few possibilities for participant observation. This was part of the motivation for organizing a workshop instead. Also, a large part of the research involved what the sociologist Harry Collins called "participant comprehension," that is, the attempt to understand what my interlocutors do. This involved studying and using their code, reading their documentation, listening to talks, attending events and workshops, and lurking in chat rooms and forums. It also involved observant participation in the form of contributing to FOSS projects, administering a server, and experimenting with salvaged hardware.

resolve it. The learning needs discussed here provide the contours of the problem situation technologists face.

This conceptual discussion is grounded in three major ethnographies of FOSS communities which I revisit in light of my own research to understand discontinuities that have emerged over the past two decades.

Recursive Deskilling

Since its inception, one of the principal purposes of FOSS has been to give users and developers greater control of the systems they work with. As the historian Christopher Tozzi put it, "[b]y retaining full access to the source code of the software that runs much of their lives, FOSS developers and users seek independence. Even if they never actually modify programs, knowing that they can do so is tremendously reassuring." More so than any substantive value, this possibility has been constitutive of FOSS.

→ Tozzi (2017, 13)

The anthropologist Chris Kelty famously argued that the people who made free software happen constituted themselves as a public in a recursive manner: as they communicated with each other, they also made and remade the infrastructure that enabled their communication in the first place. This changed not much later; an often referenced plea by Benjamin Mako Hill observed that many FOSS projects relied on nonfree tools to create software. This trend has since encompassed all aspects of FOSS. Nowadays, this is a common, perhaps even default situation for FOSS projects: their code repository is hosted on GitHub (a Microsoft-owned service), they use GitHub Actions for continuous integration testing (ditto), and developers talk to each other and to users on Discord (a closed-source, commercial platform). JavaScript developers, who produce the bulk of FOSS code these days, depend heavily on NPM, a proprietary package manager owned by a GitHub subsidiary. This is to say nothing about the use of code generation tools like Anthropic's Claude Code. Though some prominent FOSS developers have embraced them, they are still contested.

Nadia Eghbal (2020) brilliantly documents the "triumph of GitHub." See also Bounegru (2023).

Some projects employ alternatives code forges, such as Codeberg or Sourcehut, and eschew relying on other proprietary tools, but they are the exception. As a norm, FOSS projects can no longer be said to be "vitally concerned with the material and practical maintenance and modification of the … means of [their] own existence."

See Software Freedom Conservancy (2026) for some discussion and an attempt to square this circle.

→ Kelty (2008, 3)

In addition to nonfree services, the work of virtually all FOSS developers is mediated by Big Tech tools. Especially when FOSS development involves the web platform, as much of it now does, the development stack inevitably has dependencies on code supplied or sponsored by Big Tech. Big Tech players have had a decisive role in shaping everything from front-end frameworks down to browser APIs and technical standards. The economist Celia Rikap explains that "successful open sourcing is standard setting in action. As their models become industry standards, Big Tech companies reinforce their leadership using open source to signal consensus in the developers' community." Additionally, FOSS code is often deployed to cloud services, creating vendor lock-in for developers and users alike. The result is that Big Tech has attained a great degree of power over the labor process of developers.

Zhang and Carpano (2023) recount how Google, through its open source Chromium project, has been able to set web standards and gain "logistical power."

→ Rikap (2024)

As developers mostly build things that fit the world that Big Tech made, they contribute to a computing monoculture in which users are acted upon than more than they act. This deskilling of users is mirrored in the deskilling of developers. As the engineer and labor scholar Mike Cooley held, "the deskiller is deskilled." Complaints about bloat and burnout reveal that many developers see the product and process of their work as being degraded under conditions of Big Tech dominance. Even DIY technologists working on FOSS are not shielded from this; as much as their industry counterparts, they can be deskilled. This does not mean that coding and software development no longer require any skill, but that much of the "craft and craftiness" is taken out of it.

→ Cooley (1987, 79)

→ Coleman (2012, ch. 3)

FOSS was meant to give users and developers more control, but the reverse is now often the case. Developers working with stacks they do not control build apps that users can barely control. Instead of a recursive public, we have recursive deskilling. Deskilling tends to breed "unbounded cynicism and revulsion," and perhaps more FOSS developers will decide that "computers were a mistake," quit programming, and take up woodworking instead. But as Johan Söderberg and Maxigas have argued, control over labor processes is never final but always remains contested. We may take the current revival of interest in the history of Luddism as a sign of such contestation. How to effectively counter the deskilling of technical practice is one of the great learning needs of the post-FOSS moment. We can understand this as a need for a reinvention of "high-tech Luddism," that is, the defense of the independence of the labor process by technical means. But working against the dominant modus operandi, the "default settings" for technical practice, is never easy, and it is exacerbated for user-facing FOSS projects seeking to enable control over everyday technologies. They have to account for hegemonic standards of usability and availability that are premised on centralized architectures and energy and resource-intensive infrastructures. These enable convenient and frictionless experiences, but they also perpetuate the cycle of deskilling.

Kelty (2013) recognized early on that such change was afoot.

→ Braverman (1974, 151)

In a notorious essay, Marlinspike (2016) unpacks some of the challenges.

Legitimation Crisis

FOSS has had an undeniable transformative impact. Thanks to big FOSS projects like GNU, Linux, and Apache, computing applications previously only available to those with military–industrial funding came within reach of ordinary tinkerers. Despite FOSS's revolutionary impact, its action repertoire was not radical. The anthropologist Gabriella Coleman argued that FOSS could best be understood through the lens of liberalism. She did not understand liberalism as a coherent political philosophy but as a variable set of commitments. These include the familiar principles of "protecting property and civil liberties, promoting individual autonomy and tolerance, securing a free press, ruling through limited government and universal law, and preserving a commitment to equal opportunity and meritocracy." In FOSS, liberalism as an ethos entails belief in the effectiveness of contractual agreements (i.e., licenses) as means of social regulation and in the ability to collectively reach informed decisions through deliberation.

→ Coleman (2012, 2)

Free and open source software licenses famously grant four freedoms, the first of which, "freedom zero," extends the right to use the

licensed code to anyone, including multi-billion-dollar enterprises. This was contentious even in 1998, when the corporate-friendly Open Source Initiative was launched. The number of contentious voices has since steadily increased. There have been various proposals for ethical or even anticapitalist licenses that impose limits on freedom zero. Others took a more business-oriented path to prevent cloud providers from "stealing" their software (i.e., offering managed hosting without compensating the creators adequately) by relicensing their projects under "source available" or "fair source" licenses. Some projects, such as GitLab, went "open core," meaning that the basic functionality of their product ("community edition") was a FOSS project, but add-ons were proprietary or source-available only.

For example the Hippocratic License (2019) and the Anti-Capitalist Software License (2020). Related discussions date back to 2007; see Kleiner (2010).

Examples include MariaDB's Business Source License (2013), the Commons Clause (2018), and the Functional Source License (2023). Perens (2020) also points in this direction.

The critique of FOSS licensing had an unsettling effect as one of the major liberal planks of the movement was put in question. The OSI was shaken up by the proliferation of licenses. However, many other FOSS insiders responded that licenses are a means to an end, not the be-all and end-all of FOSS as a movement. Other issues shook up the FOSS world more profoundly as the debate moved from distribution and licensing to production and maintenance: Who could be involved in creating it, and who carried the burden of maintaining it?

See Lemmer-Webber (2020), Rosenzweig (2021) and, more broadly, Ehmke (2025).

The first question set in motion a long and ongoing crisis around inclusivity in the movement which occasioned not only cultural introspection but also institutional action. Seeking to redress issues around diversity and inclusion, institutions like the Software Freedom Conservancy and the Ada Initiative set up outreach programs to increase representation of women and other minorities in FOSS communities. At the same time, other stalwart institutions failed to respond adequately. Foremost among these was the Free Software Foundation where, despite numerous credible accusations against him, including accusations of a pattern of wanton misogyny, FSF founder Richard Stallman remained in leadership. This resulted in a widely-supported call for the removal of Stallman and the entire FSF board. The lack of credible response lost the FSF almost all legitimacy.

For a reconstruction, see Schneider (2021).

See DeVault (2023) for a widely circulated assessment. Governance changes since then are likely "too little, too late."

Numerous communities and individual projects adopted codes of conduct, such as the Contributor Covenant. First released by Coraline Ada Ehmke in 2014, adoption picked up in 2016 when GitHub promoted its use in projects for anti-harassment purposes. The choice to adopt a code of conduct often met with backlash from other movement insiders, something I will take up in the next section. In this context, what is important is that the existence of these codes gave community members an avenue to put governance issues on the agenda. For some FOSS communities, this significantly broadened the scope of what could be debated. In some cases, existing charters and structures proved incapable of moderating and adequately responding to debates that touched on questions of social justice within the project, the project's relationship with the military–industrial complex, or the nature of the project leadership itself. This became the biggest challenge to the implicit liberalism of FOSS, which trusted in deliberation—text heaped upon text in mailing lists, discussion forums, bug trackers, and chat rooms—to sort out collective decision-making. The shortcomings of this discursive mode were now apparent,

On the earlier history of codes of conduct, see Snelting (2020).

See Li et al. (2021). An instructive case study could be made of the NixOS project in 2024.

especially to those who were systematically sidelined and disadvantaged by political disavowal and its supposed "neutrality."

The second question—the one concerning maintenance—arose because several FOSS projects became so widely deployed that vast swaths of the internet, and thus also the digital economy, came to depend on them. One popular project, the GNU Bourne Again Shell, was found to have a security vulnerability (codenamed Shellshock) in 2014, fast on the heals of a high-profile bug in the OpenSSL library (codenamed Heartbleed) becoming known earlier that year. Both projects turned out to be mostly unpaid solo efforts. These and other FOSS projects were also essentially public goods, akin to roads and bridges. Because they were so widely relied upon by countless internet-facing services, companies in the Core Infrastructure Initiative as well as foundations and government agencies around the world attempted to turn these unruly projects into stable links in a "software supply chain." In further shifting the focus from cooperative work to creating value for corporations and the state, the resulting "maintenance turn" drew further attention to problems related to governance in FOSS.

On the rise of code reuse and the growth of software dependencies, see Cox (2019).

O'Neil et al. (2020) discuss legitimation problems that arise when FOSS projects cooperate with commercial players.

The implicit liberalism of FOSS, it increasingly appears, is not up to the challenge of addressing "unresolved steering problems" that manifest as disaffection and protracted conflict over goals, norms, and processes. A second learning need of the post-FOSS moment, then, is to figure out forms of governance that can face up to this challenge. Technologists must find ways to reconcile their identities and goals with developments, both internal and external, that call their legitimacy into question.

→ Habermas (1988, 4)

Learning Blockages

FOSS is not just a way of making software or a form of organizing a project or community. Crucially, it is also steeped in a set of norms that are closely tied up with the practices of FOSS hackers.

Steven Levy provided some of the deeper origin story of FOSS in his *Hackers: Heroes of the Computer Revolution*. He traced the beginnings of hacking to the Tech Model Railroad Club at the Massachusetts Institute of Technology (MIT) in the final years of the 1950s. Though it was still several years before TMRC got its own computer, its members were eager to play with anything they could get their hands on, the more technologically complex, the better. In part thanks to MIT's amicable relationship with the military–industrial complex, there were always plenty of toys around. In their explorations of switches and systems, these early hackers developed an ethic—the famous hacker ethic.

Levy noted that this ethic was "not so much debated and discussed as silently agreed upon," in part owing to the habits club members formed in their practical explorations of systems, both electromechanical and computational. Like Kelty and Coleman after him, Levy stressed the mutually constitutive link between hackers' practices and norms. Levy renders what he finds to be the tacit "tenets of hackerism" explicit.

1. Access to computers—and anything which might teach you something about the way the world works—should

be unlimited and total. Always yield to the Hands-On Imperative!
2. All information should be free.
3. Mistrust authority—promote decentralization.
4. Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, or position.
5. You can create art and beauty on a computer.
6. Computers can change your life for the better.

→ Levy (1984, ch. 2)

Despite forty years having passed, his formulation of the hacker ethic still circulates widely and remains a keystone even for those wanting to develop a new hacker ethic. This ethic is more permanent than a fashion but more malleable than tradition. Without being codified or institutionalized, it is normatively demanding and leaves those who deviate from it open to criticism from fellow hackers. There are multiple "genres" of hacking that revolve around this normative core.

Critics have pointed out that Levy ignored the significant European history of hacking, but even the venerable Chaos Computer Club, founded in West Germany in 1981, draws on Levy's formulation of the hacker ethic in its work.

These norms have given form to hackerdom and, by extension, FOSS. In the course of the controversies around FOSS communities' lack of inclusivity mentioned above, community members in search of systemic reasons started questioning some of these foundational tenets. Such moments of controversy can destabilize the "ground of self-evidence" and provide a window to examine assumptions and expectations that otherwise remain unquestioned. In the best case, this can spur a collective learning process; in the worst, it can lay bare blockages that stand in the way of learning.

→ Jaeggi (2018, 80)

Science and technology scholar Christina Dunbar-Hester studied open-technology cultures and their reflexive efforts to "fix" their issues with inclusivity. Through long-term ethnographic study of initiatives connected to the Python programming language (among others), Dunbar-Hester saw up-close what is involved in reforming or supplementing existing practices. Diversity advocacy is one such practice, usually taking the form of introducing technical skills, such as programming and soldering, to a more diverse group of people, especially women and racial minorities. Overwhelmingly, such efforts wound up recapitulating established practices from engineering workplaces that are ill-suited to address structural inequities. Instead, they individualize the issue, putting the onus on the individual to level up their skills to stand a chance at inclusion in technical settings.

Some of Dunbar-Hester's interlocutors recognized that inclusion had to be about more than access to technical skills. They pushed beyond efforts to bring more women and people of color into the fold, and instead tried to transform spaces of technical work themselves. They attempted "to queer or attenuate normative masculinity" and "decenter masculinity without replacing it with normative femininity" by, for instance, introducing glitter and other colorful elements. While this may seem spurious and overly concerned with surface phenomena, it can be read as an attempt to question the underlying nexus of practices that constitutes hacking.

→ Dunbar-Hester (2020, 202)

Historian Nathan Ensmenger (2010, 236–240) has shown that, as programmers developed a professional identity, rose in status, and became identified with management, they also became more male.

However, more than anything, Dunbar-Hester's ethnography shows the difficulty of introducing new practices into an established form of life. The coupling of FOSS with hacker cultures, practices from engineering and industry, and normative geek cultures results in

learning blockages. Many hackers appear unable to see the problems that stem from their modus operandi. Dunbar-Hester sees some potential in broadening the genealogy of hacking to include feminist hacking and expanding what counts as hacking to include "infrastructural care work." Though I am sympathetic to this, I have noticed that, over the past decade, many feminist hackerspaces petered out. Websites grew stale, wikis were archived, domain names weren't renewed, and people moved on. In the course of my fieldwork, I heard one activist say about their organizing work that "We used to say 'hacker'" to refer to the kinds of people they were working with in their abolitionist work. Why "used to," I asked. They responded that "plural forms of practice that are not all hacking" are needed, and that they wanted to "de-privilege" the need for technical knowledge, because "it isn't the only kind of knowledge we need." Like in the case of the interventions Dunbar-Hester discusses, queer feminism was a starting point for rethinking the boundaries of hacking, though in this case the conclusion was that it was necessary to move beyond them entirely. Similarly, many of the spaces I visited in the course of my research have defied easy classification. While I initially thought of them as hackerspaces or hacklabs, I learned when visiting them that the organizers rarely felt comfortable with these labels. Collective spaces, cultural centers, even co-working spaces were seen as more fitting—though none of these was fully satisfying. This hints at the interstitial character of these spaces and the variety of practices that were performed there—and at the desire to decenter "hacking" among these practices.

After serving as a touchstone for many, including DIY technologists and activists, increasingly these communities see the hacker form of life as uninhabitable. Though the reasons for this vary, a representative theme is the "FOSS bro" phenomenon. This epithet names the know-it-all self-assuredness that stems from many hackers' conviction of their superior mastery over computer systems. On social media, the FOSS bro appears as a "reply guy" who is quick to correct others' mistakes—especially if those others are not themselves white men. When called out for their uninvited locutions, they vociferously deny that anything other than technical know-how is in play.

Levy's formulation of the hacker ethic stated that hackers should be judged solely on the basis of their hacking and not (nontechnical) "bogus criteria." Know-how and the "hands-on imperative" are the only warrants hackers need for their actions. Accordingly, actions deserving recognition would receive that recognition solely on the basis of the achievements they make possible. This assumption of meritocracy has stood in the way of including a more diverse set of people in FOSS production. That is not only because of the actions it warrants, but also because of what it forecloses: reflection on the shortcomings of their practices. This may explain why, in the experience of some, hacking has become stagnant, even regressive. This is certainly what explains much of the backlash against codes of conduct and diversity initiatives discussed above, which has been amplified by prominent representatives of multiple FOSS communities. Such backlash is often justified by a desire to "keep politics out of software," a comment (and distinction) that can shut down all critical reflection.

See Ehmke (2015). The online magazine *Model View Culture* was an important venue questioning tech's supposed meritocracy. Earlier, Nafus (2011) provided a similar analysis.

The learning need, in this case, is a second-order need: overcoming learning blockages requires learning to learn. The resources to address the first-order crises discussed above are blocked as long as FOSS hackers dwell in blissful self-assuredness that the only warrant they require is technical mastery. Taking a more global view of hacking, we can see that it can still offer some tools to imagine alternative futures. While hacker culture is often an impediment to learning, it can also still provide a scaffold for learning.

See Chan (2014), Lindtner (2020), Beltrán (2023) and Murillo (2025) for perspectives on FOSS and hacking from the south.

Three Communities of Practice

I set out to understand how technologists address the learning needs suggested by these three concepts. The best way to do so is to look at how different communities of practice respond to the technical, political, and cultural problems of the post-FOSS moment.

My use of the term communities of practice is broadly inspired by the practice approach in the social sciences, and more immediately by the anthropology of situated learning. It refers to a collectivity of people, usually without any formal organization, who are held together by a common practice.

See also Valk (2025) for a particularly inspiring use of the concept.

Each of these practices can fruitfully be labeled a critical technical practice. Phil Agre, who coined the term, was an AI researcher and hacker at MIT who was quoted in the original *Hacker Dictionary*, but who later became a sociologist. Like him, many critical technologists —those who engage in a critical technical practice—tend to be located at the interstices between two worlds. The concept of critical technical practice has traveled significantly and was taken up in the arts and humanities where it underwent pluralization.

This kind of social location between, for instance, the arts and technical development, enables a form of reflexivity often absent from technical work. "A critical technical practice rethinks its own premises, revalues its own methods, and reconsiders its own concepts … It is not concerned with destruction but with reinvention." Technical fields, Agre finds, are often impervious to what may be called external critique, that is, critique formulated in terms that it does not share. The overriding concern "is whether it produces *solutions* to *problems*." Somebody making a philosophical critique may put their finger on something going wrong and may even get a sympathetic hearing from some practitioners. However, aside from occasioning rare individual awakenings, their critique will not precipitate a collective sense of crisis or lead to lasting transformations in the field. Such crisis and transformation can only be brought about from the inside, by working on the same terms as the practitioners.

→ Agre (1997, 27)

→ Agre (1997, 13)

All three communities of practice discussed in this report can be seen as using *code as critique* (using a broad understanding of "code"). They are working within technical practice to transform it from within. The degree of organization and coherence differs between the three communities of practice, and it is fair to ask whether the fraught sociological concept of "community" is applicable at all. Learning theorist Yrjö Engeström has suggested that, to account for the weakly bounded forms of organization that, among others, characterize FOSS communities, an analogy with mycorrhizae—networks of fungal roots—may

See Healy and Schussman (2003) for some early sociological reservations about characterizing FOSS as a "community."

be helpful. They do not have strict membership criteria, but their members can be identified; they are growing, yet they have a stable structure. They are unbounded but not indefinite. While I stick with the concept of communities of practice, I take it to include such loosely bounded formations.

I selected these three communities because in each there is a degree of "post-FOSS" sentiment even while their practices relate to different parts of today's digital infrastructure. That made these three cases particularly promising for generating new insights.

Permacomputing

Permacomputing arose from the confluence of existential concerns and a minimalist computer art movement. The existential concerns can best be summarized with reference to two fields of study, collapse informatics and post-collapse technology, that emerged in the 2010s. Technologists in these fields asked how to design systems for a future of scarcity. This included designs for a "civilizational bootstrapper," that is, a minimal toolchain to get basic computing systems up and running after a cataclysmic event.

Virgil Dupras (2020), the developer of CollapseOS, expects "our global supply chain to collapse before we reach 2030," after which "we won't be able to produce most of our electronics."

The computer art movement in question is the demoscene. Members of the demoscene, mainly based in northern Europe, get together at festivals called demoparties to show off their latest "demos"—pieces of computer art built within very tight constraints (such as file size limits). In 2020, a member of the demoscene called Viznut took up the question of "how to give computers a meaningful and sustainable place in a human civilization that has a meaningful and sustainable place in the planetary biosphere." Other artists and technologists began taking up this line of inquiry.

See Silvast and Reunanen (2014) for a historical account of the scene since its beginnings in the 1980s.

→ Heikkilä (2020)

In the years since then, many others—technologists, writers, artists and researchers among them—have adopted this concept, connecting their practices to it and expanding on Viznut's writings in glossaries and wikis. This tight interweaving of technical, existential and artistic questions is discernible in all practices related to permacomputing. They are equally concerned with matters of aesthetics and how to wean contemporary artistic practice off its computational maximalism as they are with ecological concerns and technical design principles.

The Permacomputing Wiki provides an introduction and an overview of affiliated projects. Valk and Heikkilä (2022) also provide a useful map.

Practitioners are careful to point out that permacomputing is not a search for the easy fix, as the link to permaculture may suggest, but an inherently political project. The concept of degrowth frequently serves as a bridge between the variety of concerns that come together in permacomputing.

Members of this community of practice gather in meetups and connect in forums, chat rooms, mailing lists, and on the fediverse. Its center of gravity is in Europe, and to the extent that it is formalized, it is often linked to the arts.

Server Collectives

Rather than having members of a group rely on commercial platforms, cloud storage providers, or software-as-a-service companies for the provision of digital services, the group can run the services it needs by itself: its own chat, its own email, its own file storage, its own social

media. This is what server collectives, sometimes also known as community servers, do. It is akin to the practice of self-hosting, but instead of being individually focused (a single user administering services for their own use), it is set up for and run by a collective.

This is a widely dispersed practice with a long history in internet time. The membership organizations of the Association for Progressive Communications, founded in 1990, could be considered early forerunners. They included organizations from Latin America as well as European, U.S. and Australian members. Nadir, a German collective that provided internet services (email accounts, mailing lists, web and server hosting) as well as an archive of political texts for activists on the autonomous left, began in 1994. Riseup Collective began in the United States five years later, closely followed by the Italian Autistici/Inventati collective. The best known example of such a collectively run service may be Indymedia, a major achievement of communication activism operating as a network of autonomous collectives from 1999 until it gradually petered out around 2010.

Glaser (2019) and Wolfson and Kanuga (2022) revisit the legacy of Indymedia.

In recent years this practice has expanded with the growth of the fediverse. Members of these federated social media services have to sign up with a specific server, or "instance." Avoiding centralization requires numerous independent instances, and several collectives have stepped up to run them. Collectives also continue to provide other services and infrastructure for their membership and a wider world of activists.

The operation and ongoing maintenance of collective services and infrastructures raises important questions around labor and governance in addition to technical challenges related to the sharing of administrative privileges. Many server collectives are feminist collectives, and in thinking and acting on the challenges posed by running their own infrastructure, they draw on feminist concepts and practices, in addition to principles like solidarity and self-management.

Decentralized Networking Technologies

Decentralization is a long-standing concern in computing. In the computer science fields of distributed computing, researchers study the challenge of completing computational tasks across multiple autonomous nodes that cannot even be guaranteed to agree on the passage of time. Protocols underlying everything from email (UUCP, SMTP) to the internet itself (TCP/IP, UDP) are decentralized because (at least in principle) anyone can add a node through which messages and packets can flow.

In the early 2000s, decentralization was associated with file-sharing services like Napster, and thus with "piracy." Following the Snowden revelations of 2013, there was increased attention for the architectures that make surveillance possible, sparking renewed interest among users and developers in decentralized options that would render dragnet surveillance by nation-state actors less feasible. The hype about cryptocurrencies during the 2010s was in large part about a libertarian desire for decentralization in the sense of being less dependent on "government money."

The community of practice I focus on is interested in decentralization for yet another reason. Members of this community look at the

partial success of the fediverse in the struggle for democratic social media and ask how we can improve on this achievement. Building on the ActivityPub protocol, federated social media systems like Mastodon have achieved a degree of decentralization. Because users are dependent on the "instance" they join, the governance choices made at the instance level can affect users in ways that are beyond their control. If, for example, their instance decides to defederate from another instance, they will be unable to interact with users of that instance. The only option to reconnect with them would be to start a new account on a different instance, losing their account history in the process. Decentralization promises to make the instance boundary less consequential. At the same time, the technology also promises to make new kinds of decentralized applications possible that go beyond the ActivityPub model of exchanging messages. The goal are online social worlds that combine safety, autonomy, and rich possibilities for interaction.

This community of practice is less dispersed than the other two. It centers on the Spritely Institute, a nonprofit based in the United States. Beyond that, it includes a small number of technologists who build on the same set of concepts from computer science research. Some are involved in an effort to develop a networking protocol on the basis of object capabilities known as OCapN.

Code as Critique

In November 2025, I hosted a workshop at the Netherlands Institute for Advanced Study. In the invitation to the workshop, I announced my intention to "bring together technologists who are responding in critical and practical ways to currently dominant norms and practices of technology development." For two days, two dozen participants from Europe, Asia and the Americas gathered in Amsterdam for discussions centered on the three communities of practice and the broader collective learning process they are engaged in. The workshop title was "Code as Critique: Relearning Technical Practice in the Ruins of Big Tech." In addition, the Critical Infrastructure Lab hosted us for an afternoon of hands-on work before the event culminated in a public keynote by Christine Lemmer-Webber, executive director of the Spritely Institute, that maxed out the venue's capacity and attracted a large online viewership.



Introducing the workshop.



We documented workshop activities and discussions using collective note-taking and a group chat. Our exchanges had an expansive scope, and what follows is an attempt to capture some of the main themes. I refrain from identifying individuals, though a partial list of participants is available in the acknowledgements. Passages marked as quotations are taken verbatim from our collective notes and are either direct quotes from our discussion or were paraphrased by a note-taker.

Taking Back Computing

A sentiment expressed throughout the workshop may be summarized with the sentence, "Computers weren't a mistake; data centers and devices were." This resonated with a similar sentiment that surfaced

in the remark that it was "refreshing to not debate AI all day long." Another participant retorted that, when we talk about the future of computing that we want in which one can have control over tech, "the AI goes overboard." In exploring the space beyond hegemonic visions for the future of computing, our discussions touched on many forms of know-how, skill and craft.

In our discussion of permacomputing, many of these had to do with what a participant called "a radically (slightly) more sustainable approach to computation." We discussed attempts to run computers and even whole data centers on mud batteries, running virtual machines on reclaimed, salvaged or recycled systems, including old Nintendo consoles, "smart" light bulbs and other "stuff that is already around," and how to not "overpower" one's website by building on "old" technologies rather than the latest frameworks. We also discussed more mundane maintenance and repair practices that can extend the life of consumer hardware, such as cleaning the fan of one's laptop to make it last longer. A participant from Latin America pointed out that reuse and "salvage computing" are "normal" where they are from. Against this background, permacomputing can feel "performative," but it is also adaptable and open enough to allow for multiple localized interpretations.

For more on mud batteries, see projects by Sunjoo Lee and the Critical Infrastructure Lab.

We discussed that there is often a gap between hardware being repairable, which is a focus of both regulation and some industry practices, and things being done to actually extend its longevity. Participants described workshops they offered or participated in where they learned about "staying with the same device for longer" or more broadly how to engage with computing "without contributing to waste." This involved not only learning practical maintenance and repair skills, but also developing an understanding of "digital infrastructure in terms of its political economy." Permacomputing also provides a space to discuss how a "playful minimalism" can serve as a counterpoint to graphically demanding videogames relying on GPU rendering and other forms of technological maximalism currently taken for granted across much of the computing world.

In this context, permacomputing also raises the question, "What do we even need computation for?" While permacomputing clearly does not imply abandoning computing altogether, it does encourage us to ask *what kind* of computing we need. In the world of agriculture, agroecology—which can be seen as a more radical cousin of permaculture—may arrive at the conclusion that what is needed is to burn down the field planted with proprietary seeds that threaten food sovereignty. Something similar is needed in computing as well to defend and expand practices of "participatory civilization making."

This also brought up the case of C.L.O.D.O., a French group that in the 1980s attacked data-processing sites. They are the subject of the documentary *Machines in Flames* (2022).

A more precise form of the question of what kinds of computing we need that we discussed was, "What would our work look like if we didn't have data centers?" The responses to this question segued into our discussion on decentralized network technologies. As a participant put it, "There are alternatives to having 'the server' that you have to depend on." Most current computing practice assumes that when we need to look up information, we do so by querying a database that lives on a single node. "This comes from mainframe culture, but it is not the only way of doing things." There may be reasons for having

an authoritative central node when data availability is a must, but it need not be the default. Instead, even applications that involve interaction and collaboration could be designed according to "local-first" principles where data primarily resides with users and is synchronized as needed in a decentralized manner.

Figuring out how to collaborate on things in this manner may require "taking ideas off the shelf from early computer science research" that were abandoned as the tech industry converged on cloud solutions. Social applications that do not require central intermediaries acting as custodians go against the default, but they are not without precedent. We discussed research on object capability systems which several participants described as providing useful concepts with a high potential payoff for those wanting to take back computing. The basic idea behind capabilities is simple, "but the implications are far-reaching." It is like learning "one weird trick" that makes things attainable. One appeal of object capabilities is that they provide mechanisms to model consent. Meaningful consent is "intentionally grounded, accountable, and revocable," and object capabilities allow for representing such a social process in a technical system. A simple example of how a capability system can provide a consent mechanism is the file-picker dialogue in a browser which allows for consenting to sharing a specific resource (and *only* that resource) with another party.

See Miller et al. (2003) for an introduction to capabilities.

For an example, see Karp et al. (2009).

Despite being relatively simple, making the case for object capability systems is difficult. It requires thinking differently, in terms of patterns and models that diverge from the default. This underlines the importance of experiential learning. As other technologists have found out when proposing novel ideas that go against the grain, "You can't tell people anything." But "once people experience something, it becomes more second nature." It is not enough to enumerate design principles; only a working prototype allows for the kind of experience that is necessary to convince others to try a new paradigm. For that reason, the dichotomy between technical and social problems can be misleading. "The very technical approaches we are talking about are meant to enable that world that we envision." That world is one in which systems don't simply *pretend* to offer protections to its users but actually render the kinds of harassment and surveillance facilitated by centralized systems impossible. This is something where both ActivityPub and ATProto (the protocol underlying BlueSky) fall short. "If you try to build your utopia out of papier-mâché, it just falls apart."

Agre (2003) argued that "architectures and institutions inevitably coevolve."

Data centers are also the standard response to the question "how do we scale?" They provide the tooling for the kinds of big deployments taken for granted in the "web 2.0" era. Venture capitalists who fund much tech development are given "huge bundles of cloud credits" and so "that's what they build around." FOSS projects like Ruby on Rails and Django, which many social applications are built on, often presuppose production deployments that use cloud solutions. Another possible response to the question of scale is "what if we don't scale—what do we gain from that?" Many everyday technologies do not require massive scale. They do not have to be exposed to the whole internet or be optimized for use by huge numbers of users. Instead, an app can be a home-cooked meal, and this kind of small-scale appli-

cation lends itself to a decentralized, local-first design. Such a design "liberates you from having to run a server."

Our discussion of server collectives picked up at this point. We began by asking what it means to run a server, as there are multiple levels of abstraction where "serverness" is located. One collective, we learned, gives out virtual machines to activists, and they then act as user-admins who set up their own services. Other collectives provide specific services, such as email. "Email is a service many are afraid of" because of the difficulties of running the service in such a way that it works reliably. "When we are afraid of a service, we outsource it." For that reason, a participant's collective decided to focus on email specifically.

This collective, like others, explicitly values having a dedicated physical server, and many collectives decide to move to one once they've gotten started with a virtual machine. Physical servers have an appeal for a variety of reasons. They can become a gathering point for the collective where collective learning takes place. They can also be defended against adversaries in a way that hosted solutions cannot. Police may coerce a hoster to take a virtual machine offline, but if they want to seize a physical server, activists can fight back. But physical servers can also pose risks, for instance when they are located in a territory where indigenous sovereignty is disputed and adversaries may cut off access.

In addition to being infrastructure or service providers, many server collectives have a pedagogical mission. They prioritize learning and teaching others how to run and maintain their own server. In their operation, server collectives seek an answer to the question, "what does it mean for people to have a community owned by a community rather than one that squats on somebody's website?" They want to help others overcome the initial threshold that can deter them from taking matters into their own hands. This enables artists, political activists and other groups that usually do not have the requisite know-how to run their own servers. These groups frequently foreground collective work (e.g., working together in a shared terminal session when performing system administration tasks), and they take a strong interest in understanding the required work of maintenance and repair as a form of care. Servers, in their understanding, are "situated" and "requiring care." At times, this means that the servers are individualized in ways that clashes with industry practice, where automated "provisioning" and "orchestration" of servers means they can be spun up and down at will. In industry speak, this individualization means treating the servers as "pets" rather than "cattle." Some server collectives like their servers to have some "personality." They name their servers and allow them to persist longer. At the same time, server collectives need to ensure that the work can be shared, that operational knowledge does not reside exclusively in individuals who feel the burden of keeping the servers running and risk burning out. Some collectives embrace some forms of automation in configuring and administering servers without fully following industry practice so "you don't have to know twenty different technology stacks."

The case of server collectives makes plain how taking back computing requires not only changing the defaults of technical practice,

For examples, see A Transversal Network of Feminist Servers and (2024).

Erin Kissane (2025) writes about Bonfire Social, a free software project that explicitly foregrounds questions of maintenance and care.

but also figuring out organizational forms and processes that can sustain alternative practices. That is the focus of the following section.

Growing Together

Technical systems require protocols and trust to scale out and up. Similarly, communities of practice have to devise and evolve adequate organizational forms. This becomes particularly pressing when money and legal liability have to be considered, "the legal person stuff that nobody likes to think about." In addition, organization can be a way to address reliance on invisible work which risks burning out individuals and imploding the community.

In permacomputing, governance was only just starting to emerge after four years of activity. Permacomputing was not formalized as a collective, and workshops and other events were organized on the initiative of individuals, not collectively. Gradually organizational questions became more pressing as activities became funded and the work of running the community became more complex. For this, "gardening communities can be a source of inspiration." The guide on Brewing Collectives authored by members of the permacomputing community likens organization to a fermentation process; it involves gradual interaction between elements, resulting in transformation. Even with increased attention to organization, it appears "hard to stretch this beyond people who know and trust each other personally."

Our discussion yielded a number of other sources of inspiration and metaphors. Community gardens furnished an example we repeatedly returned to. They do not have to provide a complete solution to the issue of agriculture and its political economy, but in their operation they can create environmental awareness. Similarly, practices like permacomputing need not be expected to supplant the entire computing mainstream. Instead of scaling up to embrace all of computing, such practices can scale *out*, gradually adding more small local groups. Some universities host Master Gardener programs that train individuals in the art and science of gardening, which they then take back into their communities, spreading the expertise in the process of setting up gardens. Technologists could play a similar role in leveling up knowledge and skills in local communities.

Public libraries were another source of inspiration we turned to. Big Tech companies took over tasks of libraries such as indexing and providing knowledge. What if libraries reciprocated and provided an alternative to corporate data centres? Since libraries are locally based (at least in places where there are local branches), one can imagine being able to knock on a door when experiencing problems with one's social media, and getting help from someone knowledgeable. In general, library and information science as a discipline as well as practices of archiving could inform a great deal of governance decisions technologists have to grapple with, yet they remain largely untapped. "That's who we should be talking to."

Watch Joanne McNeil and Darius Kazemi discuss the question Why Trust A Corporation To Do A Library's Job? (2021).

If such a public form of organization is out of reach (after all, "institutions may not be aligned with our best interests"), associational culture could provide another source of inspiration. The German Chaos Computer Club, for instance, is organized as a federation of local hackerspaces, each organized as a *Verein* (association), drawing

on deeply ingrained cultural scripts of voluntary organization. The collective running Codeberg is also organized in this way. Importantly, by having treasurers and bylaws, they do not put off the question of how to deal with money and maintain trust around shared resources. As such, this form of organization could also address the concern that permacomputing and other practices fetishize the small scale and shy away from organizational questions for aesthetic reasons. This could condemn a group to "always playing whack-a-mole with trust," face profound doubts regarding their legitimation, and ultimately fail to be a social force.

At the same time, the goal of any organization will also depend on context, such as the opportunity structures and concrete threats that groups face. In Europe, groups may have to consider what it would mean to accept "dirty money" from European Commission digital sovereignty initiatives, while in Latin America, activists may consider local security forces a greater threat than a Big Tech player like Microsoft. A collective that is involved in Gaza solidarity work may deal with a more urgent set of threats stemming from government repression than one focusing primarily on local needs. It is important to ask what concentrations of power exist and who is in a position to decide to exclude and censor people.

Our discussion of decentralized network technologies added additional points about organization and governance. We spoke about how ActivityPub, the protocol underlying the fediverse, can result in pressure on individual instances to settle moderation disputes. Often such disputes are experienced as "drama," since different communities have different moderation needs, and especially the presence of bad-faith actors can bring these to the fore. More than anything, this is what makes it hard to run a fediverse server. For a small community, the rules are self-explanatory; for a bigger one, the rules have to be sorted out first. Given these pressures, the extent to which the current system does work far exceeds expectations.

Kissane and Kazemi (2024) present a deep dive into such governance issues and the "federated diplomacy" they require.

Darius Kazemi (2019) advocates for running small instances on the scale of a house party.

Relieving some of these pressures would entail providing mechanisms for consent. An important insight in the context of the governance discussion is that a technical system can never fully model consent. "The real consent stuff happens outside of the system," but the system can at least provide mechanisms to get the system closer to consent. In such a system, there is greater leeway for individual decisions that can be expressed in an object capability system as described above. But governance decisions can also be delegated to others using the same capability system.

Unsurprisingly, server collectives pay a great deal of explicit attention to issues of governance. In fact, something only becomes a server collective when it is governed collectively. A small group may be able to get away with having a system of tacit rules, but starting at a certain threshold they have to be made explicit. It is important that those who use a shared server follow the rules, "but the rules don't matter nearly as much as the conflict resolution of social dilemmas." This is work that requires training and certainly cannot be automated. Community-building is always the hard part.

In a similar vein, one participant shared that "making priorities within a collective and making sure they are clear is more important

than the system being up." In their case, the collective had decided to put security first, communication second, and availability third. In a more community-oriented setting, communication may be prioritized more, and a stable communication cadence becomes important. A related consideration is that those running the server may have different things they care about than the community.

When a collective has a physical server, as is the case for many feminist collectives, its physicality can impact collective governance processes. It becomes less about providing a specific service, and more about "changing the conditions under which the services are offered." "It becomes more like a mutual aid practice."

Collectively run infrastructure can also be looked at through the lens of socio-economics and common-pool resources. Again, the comparison to a community garden proved instructive, since community gardens are a common pool and require good management to endure. There are proven design principles that collectives can draw on when dealing with issues that arise when they provide servers or services. For instance, a collective providing email services to its members may run into storage constraints when users archive all their old email, including all attachments, on the server. A graduated system of sanctions can ensure that the collective resource is not spoiled by a few who overuse it. This need not be heavy-handed; in fact, a visualization of their storage use was enough to get the users of a collective server to delete twenty gigabytes worth of attachments.

Such issues can be exacerbated in cases where collectively run services have to interface with other systems that work in an extractive manner. For instance, email is presupposed by many commercial services who send out two-factor authentication codes. In principle, email is not difficult to run and even tolerant to outages and downtime. But the stakes may change when it turns into a utility that decides whether someone can access their bank account, increasing pressure on those who maintain the service. This once again raises the labor aspect and the reliance on often invisible work that can jeopardize collective projects.

A recurring theme in our discussion was that governance work is the most difficult and also the most consequential work. This is especially apparent when looking at practices over the long term. An added challenge is that it can be hard to think on a time scale of decades. However, without thinking about the longer term, it is impossible to ensure social grounding. "If you try to design technology without a social grounding, you build fascist technology."

The Importance of Vibes

In the account of a workshop participant, the most important thing that permacomputing achieved was "taking the hackers out of the basement." This made a decisive difference with regard to identity and belonging in spaces of computational experimentation. "It was the first time I met hackers that were welcoming to people that weren't hackers themselves. Before I always immediately felt excluded or unwelcome. … I am not a hacker, I don't code, but I have the audacity to try to make a place." The shift "out of the basement" has helped to reconnect issues of computing with "the physicality of life."

Another participant made a similar observation about civic hackathons in Taiwan. There, it did not matter whether one knew how to code or not. The hackathons were very accessible and served to "match people who have technical expertise with those with ideas." Free food and an inclusive atmosphere attracted a wide range of participants, and as a result the hackathons were able to turn into a broad movement.

Even so, discussion in permacomputing circles "can become very technical." Despite feeling an affinity with technical topics, another participant said that "I am not part of those conversations. I am engaged more in the community-building and civilization-building aspects." These participants were drawn to permacomputing because "it resonated with my feelings towards technology" and because they were "questioning our use of technology."

This approach was reflected in the practical ways in which people participate in permacomputing. Members of the community offered a workshop to teach people about editing a wiki to make the concept more accessible to a wider audience. They were mindful to avoid the impression that anyone has particular expertise or that there is a center or a periphery within their community of practice, a term that they also employ as a self-description. The goal is to keep permacomputing as an "open space" that is "accessible for a wide range of publics in a non-hierarchical way." This connects to the wariness about scaling up the community mentioned above. "We want to grow but we don't want to scale," because increased scale would risk destroying the carefully cultivated sense of community—its *vibe*.

This discussion and its particular emphasis on inclusion, identity and belonging brought up the question *what else* permacomputing is aside from "vibes"? There were some reservations about this emphasis, since "in the worst case it turns it into 'just' an art project: 'Europeans doing their little art.'" Further, insisting on small scale and ethical purity may limit the political role permacomputing can play. At the same time, participants agreed that even if a significant extent of permacomputing or other practices were to be about vibes, that would not be a basis for dismissing them. An effective and sustainable response to the harassment, exclusion and censorship facilitated by currently dominant technologies requires cultural work, and "most of culture is vibes."

For that reason, our discussion of decentralized network technologies, while undeniably more technical in focus, also centered partially on vibes. The motivation for developing these technologies is grounded in specific episodes of harassment of queer people and people of color by right-wing trolls in the early years of the fediverse. Their design is based on a feminist understanding of consent that has been informed by those experiences.

Server collectives cannot operate sustainably without clear priorities and an understanding of what users of its services care about (and how that may differ from what the collective cares about). Dealing with the kinds of governance issues discussed above also requires setting the right vibe. "It's much easier coming from a non-technical background and knowing about community-building than

vice versa." It is easier to teach the basics of system administration than it is to teach interpersonal communication.

More broadly, given the location of many critical technologists between tech and the world of the arts, we discussed how the arts can help foster a different approach to computing. They do so by providing a space that is "inclusive and playful" and allowing for "experiments in collaboration with others." An artistic workshop allows a lot of liberties, and may introduce participants to the idea of logging into a server and running commands there. In the course of such an artistic workshop, participants can experience computing as empowering, "as a cultural and social activity rather than a dry thing."

There are also pitfalls, too, to locating one's work in the arts. Framing oneself as working "at the intersection of art and technology" can be a self-branding move, subtly creating a new form of elitism that throws up new barriers rather than inviting people in. Participation can also have a negative side. The publics being invited into artistic workshops may eventually begin to feel fatigue. There is also a risk that the audience has little overlap with those most negatively affected by automated systems, since they are often not the people that come to art events.

Stevenson and Valente Pinto (2024, 3772) ask "whether a critical mode of distinction is simply replacing an outmoded hacker masculinity."

Building a Neighborhood

What potential is there, then, of building bridges from the communities of practice to other publics or groups? As a participant in the workshop, at one point I found myself reminded of Langdon Winner's critique of the 1970s appropriate technology movement. Winner remarked that participants in the movement seemed to believe that it was sufficient to "build the better mousetrap," and that this would automatically result in a market for their ideas. As the discussion progressed, we kept coming back to the question of how to "raise awareness" and "spread knowledge" beyond the world of critical technologists. Why should others care and embrace a different approach to computing?

Appropriate technology and other movements from the past have also inspired permacomputing, as Valk (2021) documents.

→ Winner (1986, ch. 4)

The concept of permacomputing has brought in "a range of publics from laypeople or hackers." What they share are similar anxieties, but also common interests that go beyond computing: sustainability, gardening, and agriculture. Although permaculture can be "a bit cultish, a small church within the broader movement of organic sustainable agriculture," it can function as an on-ramp to permacomputing. People that are already interested in permaculture or agroecology may see resonances between what they know and the different approach to computing suggested by permacomputing. In fact, some permacomputing groups started in actual gardening, "by scratching their own itch." This can also help to give the practice local roots, as it can connect to already existing community gardens. Like community gardens or bike kitchens, permacomputing could then serve to "distribute agency in a community" by spreading skills and expertise.

The aforementioned vibe in permacomputing helps to "keep the form open to newcomers, making sure people feel welcome." Participating in permacomputing can take many different forms, and may even include a refusal to engage with computing. "Sometimes people propose not to do anything with computers for the day." Even so, since

computing is an important "aspect of contemporary civilization," it is a topic that many populations can relate to. "Queer marginalized folks often find their refuge in digital spaces which depend on computation." For them, the question of how to maintain representation is crucial. A challenge in making the case to them can be that they may actually prefer "somebody anonymous and faceless at Microsoft" running their services than a person they have a real-life relationship with. But as phenomena like the "cozy web" show, there are people who value choosing their community and sharing in intentional ways. They may recognize the value of inhabiting digital space in ways that correspond more closely to how they inhabit physical space, where it is easier to maintain contextual integrity. For them, the question of how to move between spaces in different communities is crucial, and the appeal of systems that do not scale may be clearer.

Decentralized network technologies respond to the need and desire to do things together in intentional ways. They challenge "the libertarian idea of the internet as inherently global," and instead allow for the subsidiary level where consent is possible. At this level, one does not have to find a one-size-fits-all answer; "the balance can be found in each community according to their needs." Still, the question remains how to get users to care. The challenge is that one needs to have feature parity with, for instance, the Microsoft cloud to entice users to migrate away. Exposing how things work through design might get people to care about the differences between incumbent technologies and alternatives. An analogy is that some consumers do not shy away from the effort to hand-grind their coffee, and they learn to appreciate the added friction. Just as there are enough people who care where their coffee is sourced from to create a market for specialty coffee, there may be enough users who consider the differences between centralized platforms and services built on decentralized protocols important. There were people who joined the fediverse early on and "put up with its growing pains." Even now, when a large Mastodon instance goes down, "people do not mind the availability issue; they know we are doing it with a mission." Users can have an understanding of and appreciation for "the human labor behind the scenes."

There are young internet users who never experienced GeoCities but who build websites on Neocities. This can keep the idea that computing can something they actively participate in alive, despite the mostly passive ways recent generations grow up with technology. Beyond generational differences, there are also class differences. Access to devices varies, and who can or cannot develop a skilled approach to computing varies accordingly. There are "pockets of enfranchised users" that continue to tinker with code and hardware, such as people setting up their own "smart homes." They have to become very skilled and engage in ongoing maintenance, otherwise "the lights literally won't turn on." Another example are artists who create visual art with Blender. For them, the boundary between using the program and extending it through programming is fluid, as manual actions can be turned into general procedures to create complex visual effects. We also should not underestimate the intrinsic joy that can come from playing around. Warnings about privacy and "internet health" can

turn people away because they sound like stern parental reminders to eat one's vegetables. Instead of scolding users, "how can we make chocolate-covered broccoli?"

Again, the arts provide promising examples. Many artists have become interested in technology because of the huge role social media plays in how they get their work out. Instagram is one's portfolio; people ask for one's handle before they ask for an email address. Because of that, some artists start taking an interest in the infrastructure behind it. The "artistic drive" to get to the bottom of things can be a powerful motor driving exploration of hidden layers of labor and technology. May artistic practices lend themselves to such a process of exploration. Experimentation with alternative publishing practices can incubate ideas that spread to other kinds of work. Live-coding (performing music and visuals with code) and other artistic formats can give people a sense of agency. Dance and other artistic forms that center embodiment can provide a means of exploring how systems work in a nonverbal, nondiscursive way, which can make them both more accessible and more enjoyable for new publics. Such playfulness is also a way of avoiding didacticism.

The arts can provide a flexible framework "to bring people into a world that normally is far away from them." This can serve not only as a way to teach, but also to build community. Artists recognize that "finding formats is one of the biggest challenges." Ideally they provide an opportunity for "deep or close listening." In such a setting, it could be possible to "rethink what is actually needed and what we expect from computing."

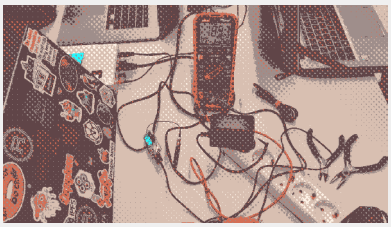
Our hands-on session provided us with an opportunity to experience for ourselves the excitement and joy that can come with approaching computing in surprising and new ways (see Box 2). We learned that it is possible to engage with computation using only our bodies by performing dance choreographies. We learned that bacteria in polluted mud taken from the bottom of Amsterdam's canals can produce an electric current, and that a sufficient amount of this mud can be enough to power an ESP32 device capable of relaying data through the æther.

A bigger question that emerged from our discussions and our experiences together was whether there is a way to keep the twin hopes of *sustainable computing* and *participatory computing* alive. To the extent that people are drawn to alternative practices, their desires often fit within the intersection between these visions.

These experiments hearken back to work by British cyberneticists Gordon Pask and Stafford Beer in the 1950s and 1960s. As Pickering (2010) puts it, they provided "sketches of another future."

Box 2: Impressions from the Hands-On Session





Discussion

In this report, I have tried to present a corrective to lingering ideas about FOSS that have reached their expiration date in the present "post-FOSS" moment. We need to retire notions such as the one that peer production is inherently democratizing or that hacking is a route to social transformation. These ideas—and other stories of open source's success—were not misguided, but they have run their course (at least as far as purported non-economic impacts are concerned). As I have tried to show, this conclusion is not one arrived at from the outside, for instance through an assessment of FOSS in light of democratic theory or an analysis of its political economy. Instead, movement insiders, particularly those DIY technologists most invested in the idea that FOSS should promote user freedom by producing convivial tools, have increasingly grown disillusioned with the movement's *doxa*. This has had an unsettling effect on the movement as a whole, even as it appears victorious.

Despite times having changed, these ideas live on, and FOSS remains invested with utopian hopes that are now unfounded. This report hopes to offer a corrective, at least in so far as suggesting a different map for where to focus such hopes. The communities of practice centered in this report may have some overlap with FOSS as traditionally understood, but they also point beyond it in several crucial ways.

The results presented in this report are the product of multi-year research using multiple forms of engagement with a diverse array of interlocutors. They are presented not as definitive answers but as "inklings" to do justice to their tentative nature. In addition to restating the key insights from this research, in the following I also point out some potential avenues for further research that I think could provide additional support for the ideas I present here, but that may also "inconvenience" some of them and demand additional clarification or nuance.

Key Insights

The practices and norms of FOSS are in crisis because they have become problematic for a key constituency: technologists who work on everyday technologies meant to expand user freedom, such as end-user applications, federated social media and other networked services serving as important infrastructures in daily life. I have suggested that three mutually compounding developments lie at the root of this crisis. *Recursive deskilling* means that the process and product of software development work has been degraded by Big Tech, and due to the hold Big Tech has on stacks and platforms, this extends to the work of many DIY technologists who historically were FOSS contributors and maintainers. The deskilling also extends to users who are left to deal with degraded (some might say "enshittified") systems. The *legitimation crisis* has called into question two of the main planks of FOSS, namely licenses and a deliberative, depoliticizing mode of social coordination. Finally, there are *learning blockages* that result from the nexus of social practices in FOSS forged in hacker culture, and these make it harder to resolve collective problems.

The Code as Critique workshop brought together members from three communities of practice, and it allowed us to explore how critical technologists are grappling with these developments.

Against the deskilling of developers and users, technologists are working to reclaim a vision of computing that is not locked into a monoculture of locked-down devices and centralized clouds. All three communities forgo the industry emphasis on "scaling" and instead emphasize minimalist, intentional and situated forms of engaging with computing. These are inherently more demanding of skillful practice since they go against industry defaults and do not fit the dominant modus operandi. They may also place higher demands on users, requiring the sharing of skills and expertise to "distribute agency."

The legitimation crisis of FOSS did not come up explicitly during the workshop. But in the many ways they reflected on issues of governance, participants revealed a vivid political imagination that points beyond the implicit liberalism informing FOSS practices. This imagination is fed by numerous examples ranging from mutual aid collectives to community gardens to public libraries which model ways in which an organization can coordinate work, handle money, and ensure ongoing operation without burning out individual contributors. Tellingly, participants did *not* mention any traditional institutional players in the FOSS ecosystem, such as the FSF, OSI or SFC. They also did not take a position on copyleft versus permissive licensing. Though their sympathies may lie with copyleft, they have no illusions about the political efficacy of licensing. There is not one organizational mechanism or form that critical technologists have moved on to. The insight that stands out to me is that they recognize the need for multiple forms of organization and are hungry to learn about and practice them.

The learning blockages that result from the legacy of hacker cultures and how to overcome them were also addressed by themes that ran through the workshop. Participants repeatedly questioned the distinction between "the technical" and "the political" that often designates necessary political discussions in DIY technology communities as "off-topic." The discussion of "vibes" that ran through the workshop gave a more specific direction to this. It showed that issues of identity, belonging and queerness are not something that are brought into technical practice from the outside, but that they animate it from the inside.

Further Research

This research has looked beyond the "GitHub generation" which is often the focus of research, and as such it broadens the perspective on who is making technology, how they do it, and what motivates them. Yet it is also undeniably centered on a bubble of sorts. As an Amsterdam-based researcher, I have greater awareness of what happens in my hometown as well as in Brussels, Rotterdam, Berlin, Vienna and other European cities with lively scenes of people working at "the intersection of art and technology." Despite my best efforts to expand my horizons by lurking in a variety of spaces, reading widely, and speaking to people from around the world, I admit to a degree of provincialism.

Further research would do well to widen the geographic scope. For instance, there is important work that points in a "post-FOSS" direction being done at the Núcleo de Tecnologia of Brazil's Homeless Workers' Movement, in the Tierra Común collective and in many other initiatives across Latin America.

Other practices are also worth studying. As more parts of the world brace for war or deal with its fallout, the existential concerns that animate permacomputing become more acute and fuel the growth of practices around mesh networking, which promises a greater degree of resilience than conventional infrastructure. It would be particularly valuable to look into collectives working with the Reticulum Network Stack and other mesh technologies. The DWeb community is another focal point of many development relating to peer-to-peer, local-first, and regenerative approaches, and it has recently broadened its scope. Malleable Systems Collective and Feeling of Computing are other communities of practice where issues around minimalism and participatory computing are explored. Once we reach a plateau in the hype cycle, collective explorations of large language model technologies may become feasible, and a community of practice may attempt to find uses of local LLMs that fit the visions of sustainable and participatory computing.

Several members of the workshop were excited about the Cables of Resistance conference hosted in Berlin in April 2026. Its manifesto called for establishing "radically democratic control over digital infrastructures" and many technologists actively participated to discuss alternative futures. Some who were there said it felt like the beginning of something.

This report has tried to provide a map; other travelers may linger at specific points and fill in blank spots as they explore the territory by going into greater depth on specific practices.

Conclusion

It is hard not to feel pessimistic about the state of digital technology at this moment. Big Tech has a stranglehold and does not even pretend to be progressive anymore. FOSS maintainers, already overwhelmed by low-effort drive-by pull requests submitted by individuals who were told to treat GitHub as their CV, are now flooded by no-effort slop submitted by coding agents. Age verification laws and other legal proposals around the world threaten to put an end to the idea that computing can be participatory and shaped by anyone other than big corporate players.

The litany could continue. But there are also reasons to feel hopeful. Workshop participants reported that they felt energized by the discussion, and there were ample signs of genuine collective learning happening. Moments of crisis hold opportunity because they make it easier to adopt new patterns of action.

If FOSS helped create this moment in technology, as many technologists believe, then "post-FOSS" practices and the cultural retooling they make possible may play a small part in devising strategies for resisting—and perhaps even undoing—it.

Acknowledgements

Much appreciation and gratitude to those who took part in the Code as Critique workshop: Anders Aarvik, Brooke, bumblefudge.com, Carolina, Christine Lemmer-Webber, Christy, Darius Kazemi, Hazel, Ivo, James McGrail, Jordi Viader Guerrero, Kars Alfrink, Letizia Chiappini, Liz Steininger, Marion Tränkle, Maxigas, Michael Stevenson, Niels ten Oever, Nurul Huda Rashid, Ola, Rodrigo Ochigame, Sejo, Tommi Marmo, Ying Tzu Lin, and those who preferred not to be named. Special thanks to the Critical Infrastructure Lab for a wonderful afternoon filled with joy and critical technical practice.

I am grateful to the Digital Infrastructure Insights Fund for enabling me to carry out this research driven by an admittedly somewhat intractable research question. Fund director Katharina Meyer was an immeasurable help and valuable interlocutor from beginning to end. It was great to be part of a funding cohort and to be able to benefit from the expertise of fellow grantees, including Erin Kissane, Darius Kazemi, and Luis Felipe Murillo.

While working on this project, I was fortunate to be a fellow at the Netherlands Institute for Advanced Study in the Humanities and Social Sciences. Thanks to everyone at NIAS, the staff, the librarians, and fellow 2024–25 fellows who supported and encouraged me—too many to name here. Thanks to Berthon Alberto, Klaartje Laan, René Biermans and Isabel Schneider for helping make my NIAS workshop happen. I also benefited from the support of many colleagues at Leiden University. You know who you are.

All uses of em dashes, the rule of three, and elegant variation are organically sourced and lovingly planted in the text without the support of LLMs.

References

Agre, Philip E. 1997. *Computation and Human Experience*. Cambridge University Press.

Agre, Philip E. 2003. "P2p and the Promise of Internet Equality." *Communications of the ACM* 46 (2): 39–42. https://doi.org/10.1145/606272.606298.

Beltrán, Héctor. 2023. *Code Work: Hacking across the US/México Techno-Borderlands*. Princeton University Press.

D. Bevilacqua. 2024. *Artists Running Data Centers*. Servus.at.

Bounegru, Liliana. 2023. "The Platformisation of Software Development: Connective Coding and Platform Vernaculars on GitHub." *Convergence: The International Journal of Research into New Media Technologies* 30 (6): 2212–32. https://doi.org/10.1177/13548565231205867.

Boy, John D. 2022. "Free Software Development as Observant Participation." In *FOSDEM '22*.

Braverman, Harry. 1974. *Labor and Monopoly Capital: The Degradation of Work in the Twentieth Century*. Monthly Review Press.

Chan, Anita Say. 2014. *Networking Peripheries: Technological Futures and the Myth of Digital Universalism*. MIT Press.

Claburn, Thomas. 2026. "Chardet Dispute Shows How AI Will Kill Software Licensing." *The Register*, March 6. https://www.theregister.com/2026/03/06/ai_kills_software_licensing/.

Coleman, E. Gabriella. 2011. "Hacker Politics and Publics." *Public Culture* 23 (3): 511–16. https://doi.org/10.1215/08992363-1336390.

Coleman, E. Gabriella. 2012. *Coding Freedom: The Ethics and Aesthetics of Hacking*. Princeton University Press.

Coleman, E. Gabriella, and Alex Golub. 2008. "Hacker Practice: Moral Genres and the Cultural Articulation of Liberalism." *Anthropological Theory* 8 (3): 255–77. https://doi.org/10.1177/1463499608093814.

Collins, Harry. 2019. *Forms of Life: The Method and Meaning of Sociology*. MIT Press.

Cooley, Mike. 1987. *Architect or Bee? The Human Price of Technology*. Hogarth.

Cotton, Ben. 2021. "What Does "Open Source" Mean in 2021?." January 25. https://funnelfiasco.com/blog/2021/01/25/what-does-open-source-mean-in-2021/.

Cox, Michael, Gwen Arnold, and Sergio Villamayor Tomás. 2010. "A Review of Design Principles for Community-Based Natural Resource Management." *Ecology and Society* 15 (4). https://doi.org/10.5751/es-03704-150438.

Cox, Russ. 2019. "Surviving Software Dependencies: Software Reuse Is Finally Here but Comes with Risks." *ACM Queue* 17 (2): 24–47. https://doi.org/10.1145/3329781.3344149.

DeVault, Drew. 2023. "The Free Software Foundation Is Dying." April 11. https://drewdevault.com/2023/04/11/2023-04-11-The-FSF-is-dying.html.

Dunbar-Hester, Christina. 2020. *Hacking Diversity: The Politics of Inclusion in Open Technology Cultures*. Princeton University Press.

Dupras, Virgil. 2020. "Collapseos – Why?.". https://collapseos.org/why.html.

Eghbal, Nadia. 2016. *Roads and Bridges: The Unseen Labor behind Our Digital Infrastructure*. Ford Foundation.

Eghbal, Nadia. 2020. *Working in Public: The Making and Maintenance of Open Source Software*. Stripe.

Ehmke, Coraline Ada. 2015. "The Dehumanizing Myth of the Meritocracy." *Model View Culture*, no. 21. https://modelviewculture.com/pieces/the-dehumanizing-myth-of-the-meritocracy.

Ehmke, Coraline Ada. 2025. *We Just Build Hammers: Stories from the Past, Present, And Future of Responsible Tech*. Apress.

Engeström, Yrjö. 2007. "From Communities of Practice to Mycorrhizae." In *Communities of Practice: Critical Perspectives*, edited by Jason Hughes, Nick Jewson, and Lorna Unwin. Routledge.

Ensmenger, Nathan L. 2010. *The Computer Boys Take over: Computers, Programmers, And the Politics of Technical Expertise*. MIT Press.

Geenen, Daniela van, Karin van Es, and Jonathan W. Y. Gray. 2023. "Pluralising Critical Technical Practice." *Convergence: The International Journal of Research into New Media Technologies* 30 (1): 7–28. https://doi.org/10.1177/13548565231192105.

Gehl, Robert W. 2025. *Move Slowly and Build Bridges: Mastodon, the Fediverse, And the Struggle for Democratic Social Media*. Oxford University Press.

Geuter, Jürgen "tante". 2025. "A Luddist Criticism of Open Source." In *Fluconf '25*. https://archive.org/details/tante-a-luddist-criticism-of-open-source.

Gillespie, Tarleton. 2018. *Custodians of the Internet: Platforms, Content Moderation, And the Hidden Decisions That Shape Social Media*. Yale University Press.

Glaser, April. 2019. "Another Network Is Possible." *Logic*, no. 8. https://logicmag.io/bodies/another-network-is-possible/.

Habermas, Jürgen. 1988. *Legitimation Crisis*. Translated by Thomas McCarthy. Polity.

Hasinoff, Amy A., and Nathan Schneider. 2022. "From Scalability to Subsidiarity in Addressing Online Harm." *Social Media + Society* 8 (3). https://doi.org/10.1177/20563051221126041.

Healy, Kieran, and Alan Schussman. 2003. "The Ecology of Open Source Software Development." January 14. https://kieranhealy.org/publications/oss-ecology/.

Heikkilä, Ville-Matias "Viznut". 2020. "Permacomputing." June 24. http://viznut.fi/files/texts-en/permacomputing.html.

Hill, Benjamin Mako. 2010. "Free Software Needs Free Tools." June 4. https://mako.cc/writing/hill-free_tools.html.

Horn, Melody. 2020. "Post-Open Source." August 30. https://www.boringcactus.com/2020/08/13/post-open-source.html.

Hoye, Mike. 2018. "The Evolution of Open." November 9. http://exple.tive.org/blarg/2018/11/09/the-evolution-of-open/.

Jaeggi, Rahel. 2018. *Critique of Forms of Life*. Translated by Ciaran Cronin. Harvard University Press.

Karp, Alan, Marc Stiegler, and Tyler Close. 2009. "Not One Click for Security?." In *SOUPS '09*. https://doi.org/10.1145/1572532.1572558.

Kazemi, Darius. 2019. "Run Your Own Social." July 8. https://runyourown.social/.

Kelty, Christopher M. 2008. *Two Bits: The Cultural Significance of Free Software*. Duke University Press.

Kelty, Christopher M. 2013. "There Is No Free Software." *Journal of Peer Production*, no. 3. https://peerproduction.net/issues/issue-3-free-software-epistemics/debate/there-is-no-free-software/.

Kissane, Erin. 2025. "Sparks Fly Up." November 11. https://www.wrecka.ge/sparks-fly-up/.

Kissane, Erin, and Darius Kazemi. 2024. *Governance on Fediverse Microblogging Servers*. Digital Infrastructure Insights Fund. https://fediverse-governance.github.io/.

Klabnik, Steve. 2019. "What Comes After Open Source?." April 2. https://steveklabnik.com/writing/what-comes-after-open-source/.

Kleiner, Dmytri. 2010. *The Telekommunist Manifesto*. Institute of Network Cultures.

Kleppmann, Martin, Adam Wiggins, Peter van Hardenberg, and Mark McGranaghan. 2019. "Local-First Software: You Own Your Data, In Spite of the Cloud." Ink & Switch, April. https://www.inkandswitch.com/essay/local-first/.

Lave, Jean, and Etienne Wenger. 1991. *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press.
Lemmer-Webber, Christine. 2020. "What Should Fit in a FOSS License?." March 9. https://dustycloud.org/blog/what-should-fit-in-a-foss-license/.
Levy, Steven. 1984. *Hackers: Heroes of the Computer Revolution*. Doubleday.
Li, Renee, Pavitthra Pandurangan, Hana Frluckaj, and Laura Dabbish. 2021. "Code of Conduct Conversations in Open Source Software Projects on GitHub." *Proceedings of the ACM on Human-Computer Interaction* 5 : 1–31. https://doi.org/10.1145/3449093.
Lindtner, Silvia M. 2020. *Prototype Nation: China and the Contested Promise of Innovation*. Princeton University Press.
Lorusso, Silvio. 2021. "The User Condition: Computer Agency and Behavior." February 12. https://theusercondition.computer/.
Malik, Maya, and Momin M. Malik. 2021. "Critical Technical Awakenings." *Journal of Social Computing* 2 (4): 365–84. https://doi.org/10.23919/JSC.2021.0035.
Mansoux, Aymeric, and Roel Roscam Abbing. 2020. "Seven Theses on the Fediverse and the Becoming of FLOSS." In *The Eternal Network: The Ends and Becomings of Network Culture*, edited by Kristoffer Gansing and Inga Luchs. Institute of Network Cultures.
Mansoux, Aymeric, Brendan Howell, Dušan Barok, and Ville-Matias Heikkilä. 2023. "Permacomputing Aesthetics: Potential and Limits of Constraints in Computational Art, Design and Culture." In *LIMITS '23*. https://doi.org/10.21428/bf6fb269.6690fc2e.
Marlinspike, Moxie. 2016. "The Ecosystem Is Moving." Signal Technology Foundation, May 10. https://signal.org/blog/the-ecosystem-is-moving/.
Milan, Stefania. 2016. "Liberated Technology: Inside Emancipatory Communication Activism." In *Civic Media: Technology, Design, Practice*, edited by Eric Gordon and Paul Mihailidis. MIT Press.
Miller, Mark S., Ka-Ping Yee, and Jonathan Shapiro. 2003. "Capability Myths Demolished.". http://zesty.ca/capmyths/.
Morningstar, Chip. 2004. "You Can't Tell People Anything." April 22. https://habitatchronicles.com/2004/04/you-cant-tell-people-anything/.
Mueller, Gavin. 2021. *Breaking Things at Work: The Luddites Were Right About Why You Hate Your Job*. Verso.
Murillo, Luis Felipe R. 2025. *Common Circuits: Hacking Alternative Technological Futures*. Stanford University Press.
Nafus, Dawn. 2011. "Patches Don't Have Gender": What Is Not Open in Open Source Software." *New Media & Society* 14 (4): 669–83. https://doi.org/10.1177/1461444811422887.
Nissenbaum, Helen. 2009. *Privacy in Context: Technology, Policy, And the Integrity of Social Life*. Stanford University Press.
O'Neil, Mathieu, Laure Muselli, Mahin Raissi, and Stefano Zacchiroli. 2020. "Open Source Has Won and Lost the War": Legitimising Commercial–Communal Hybridisation in a FOSS Project." *New Media & Society* 23 (5): 1157–80. https://doi.org/10.1177/1461444820907022.

Olson, Jes. 2022. "Drones Run Linux: The Free Software Movement Isn't Enough." September 16. https://j3s.sh/thought/drones-run-linux-free-software-isnt-enough.html.

Ostrom, Elinor. 1990. *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press.

Parrish, Allison. 2016. "Programming Is Forgetting: Toward a New Hacker Ethic." October 7. http://opentranscripts.org/transcript/programming-forgetting-new-hacker-ethic/.

Perens, Bruce. 2020. "What Comes After Open Source?." August 24. https://perens.com/2020/08/24/what-comes-after-open-source/.

Pickering, Andrew. 2010. *The Cybernetic Brain: Sketches of Another Future*. University of Chicago Press.

Ridgway, Renée. 2026. "From Free Software to Open Source: Traversing the Terminologies, Values and Ethics of an Open Knowledge Infrastructure." In *The Politics of Open Infrastructures: Power, Governance, And Justice in Digital Knowledge Practices*, edited by Katja Mayer, Astrid Mager, and Renée Ridgway. Open Book Publishers. https://doi.org/10.11647/OBP.0528.02.

Rikap, Celia. 2024. *Dynamics of Corporate Governance Beyond Ownership in AI*. Common Wealth. https://www.common-wealth.org/publications/dynamics-of-corporate-governance-beyond-ownership-in-ai.

Rosenzweig, Alyssa. 2021. "Software Freedom Isn't About Licenses—It's About Power." March 28. https://rosenzweig.io/blog/software-freedom-isnt-about-licenses-its-about-power.html.

T. R. Schatzki, K. Knorr Cetina, E. von Savigny. 2001. *The Practice Turn in Contemporary Theory*. Routledge.

Schneider, Nathan. 2021. "The Tyranny of Openness: What Happened to Peer Production?." *Feminist Media Studies* 22 (6): 1411–28. https://doi.org/10.1080/14680777.2021.1890183.

Schrape, Jan-Felix. 2019. "Open-Source Projects as Incubators of Innovation: From Niche Phenomenon to Integral Part of the Industry." *Convergence: The International Journal of Research into New Media Technologies* 25 (3): 409–27. https://doi.org/10.1177/1354856517735795.

Silvast, Antti, and Markku Reunanen. 2014. "Multiple Users, Diverse Users: Appropriation of Personal Computers by Demoscene Hackers." In *Hacking Europe: From Computer Cultures to Demoscenes*, edited by Gerard Alberts and Ruth Oldenziel. Springer. https://doi.org/10.1007/978-1-4471-5493-8_7.

Sloan, Robin. 2020. "An App Can Be a Home-Cooked Meal." February. https://www.robinsloan.com/notes/home-cooked-app/.

Snelting, Femke. 2020. "Codes of Conduct: Transforming Shared Values into Daily Practice." In *The Beautiful Warriors: Technofeminist Praxis in the Twenty-First Century*, edited by Cornelia Sollfrank. Autonomedia.

Söderberg, Johan, and Maxigas. 2022. *Resistance to the Current: The Dialectics of Hacking*. MIT Press.

Software Freedom Conservancy. 2026. "Facing LLM-gen-AI in FOSS." June 18. https://sfconservancy.org/llm-gen-ai/.

Stevenson, Michael, and Carolina Valente Pinto. 2024. "Distinction and Alternative Tech: Exploring the Techno-Critical Disposition."

New Media & Society 27 (7): 3770–87. https://doi.org/10.1177/14614448241239579.
Swidler, Ann. 1986. "Culture in Action: Symbols and Strategies." *American Sociological Review* 51 (2): 273–86. https://doi.org/10.2307/2095521.
Tomlinson, Bill, Eli Blevis, Bonnie Nardi, Donald J. Patterson, M. Six Silberman, and Yue Pan. 2013. "Collapse Informatics and Practice: Theory, Method, And Design." *ACM Transactions on Computer-Human Interaction* 20 (4): 1–26. https://doi.org/10.1145/2493431.
Tozzi, Christopher. 2017. *For Fun and Profit: A History of the Free and Open Source Software Revolution*. MIT Press.
Valk, Marloes de. 2021. "A Pluriverse of Local Worlds: A Review of Computing within Limits Related Terminology and Practices." In *LIMITS '21*. https://doi.org/10.21428/bf6fb269.1e37d8be.
Valk, Marloes de. 2025. "The Image at the End of the World: Communities of Practice Redefining Technology on a Damaged Earth." Doctoral dissertation, London South Bank University.
Valk, Marloes de, and Ville-Matias "Viznut" Heikkilä. 2022. "Permacomputing." *Counter-n*, ahead of print, December 7. https://doi.org/10.18452/25609.
Villa, Luis. 2016. "Free as In...?." March 23. https://lu.is/2016/03/free-as-in-my-libreplanet-2016-talk/.
Winner, Langdon. 1986. *The Whale and the Reactor: A Search for Limits in an Age of High Technology*. University of Chicago Press.
Wolfson, Todd, and Malav Kanuga. 2022. "When We Were the Media." *Logic*, no. 18. https://logicmag.io/pivot/when-we-were-the-media/.
Zhang, Bolun, and Davide Carpano. 2023. "Chromium as a Tool of Logistical Power: A Material Political Economy of Open-Source." *Big Data & Society* 10 (1). https://doi.org/10.1177/20539517231182399.