Digital Geography 7 (2024) 100101

Contents lists available at ScienceDirect

Digital Geography and Society

journal homepage: www.sciencedirect.com/journal/digital-geography-and-society

From FOSS to profit: Digital spatial technologies and the mode of production

Dillon Mahmoudi [a,*], Jim Thatcher [b], Laura Beltz Imaoka [c], David O'Sullivan [d]

[a] Geography and Environmental Systems, University of Maryland, Baltimore County, Baltimore, United States of America
[b] College of Earth, Ocean and Atmospheric Sciences, Oregon State University, Corvallis, United States of America
[c] Critical Media Studies, University of Texas at Dallas, Dallas, United States of America
[d] School of Environment, University of Auckland and Geospatial Stuff, Auckland, New Zealand

ARTICLE INFO

Keywords:
Mode of production
GeoJSON
Shapefile
Dead labor
Open source

ABSTRACT

Existing literature has scrutinized the impact of geospatial technologies from various angles. This article adopts a heterodox vantage point—the mode of production—to illuminate the intricate power dynamics woven into the fabric of these technologies. By focusing on the mode of production, we meticulously demonstrate how ostensibly novel digital technologies and geospatial data formats wield power within social relations of production. Responding to calls to scrutinize the political economy of spatial technologies and map-making tools, we aim to unravel the underpinning social relations, software development techniques, and technologies that shape file formats like GeoJSON and Esri Shapefile. By tracing the historical evolution of these formats, the article reveals how digital labor, both voluntary and expropriated, shapes the landscapes of profit-driven technology firms. The rise of open standards is not a departure from for-profit motives but rather a manifestation of the confluence of free, open, and for-profit. Ultimately, we argue that the intricate connections between digital technologies, geography, and capitalist structures enroll seemingly independent FOSS products into broader systems of capital accumulation. These findings highlight the far-reaching impact of geospatial technologies and their role in perpetuating and reshaping capitalist dynamics.

1. Introduction

Within Geography and cognate disciplines, "the digital" writ large continues its rapid ascent as "both object and subject of geographical inquiry" (Ash et al., 2016). On many levels, the growing interest in the study of, with, and through digital technologies and data seems obvious. From the astronomical valuations of technology giants like Google, Apple, and Amazon (Swartz, 2020) to the growing national and international security focus on digital technologies such as artificial intelligence (Hälterlein, 2021; Paglen, 2009), and the intimate ways digital technologies mediate daily life through Zoom calls and Tinder matches (Miles, 2017; Richardson, 2018), "the digital" is a central feature of modern society and an essential lens for its examination. Both academic and popular presses have explored the intersections of digital technologies and society from various perspectives. These works demonstrate, for example, how cities already perform many of the purportedly novel functions of smart cities without being labeled as "smart" (Shelton et al., 2015), how technology firms continually reimagine and rebrand themselves as technical solutions to social issues (McNeill, 2015), the asymmetric relations between data creators and data owners (Thatcher et al., 2016), the dynamics of cryptocurrency markets (Klein, 2021), and the ecological impacts of digital technologies (Lally et al., 2019; Mahmoudi et al., 2020; Turnbull et al., 2023). While these bodies of work make significant contributions to the disentanglement of winners and losers within the proliferation of digital technologies, with notable exceptions (Fuchs, 2014; Thatcher, 2017; Wayne, 2004), they do not engage digital technologies through the lens of the *mode of production*.

By focusing on the mode of production, this article (re)centers how ostensibly new digital technologies and geospatial data formats entrench and distribute power in the social relations of production. In doing so, we respond to calls from O'Sullivan (2006) and Thatcher et al. (2016) to examine the political economy of spatial technologies and of, simply, the technologies involved in making maps. It becomes not only important to implicate technologies and digital labor, but also the social

---

* Corresponding author.
E-mail addresses: dillonm@umbc.edu (D. Mahmoudi), jim.thatcher@oregonstate.edu (J. Thatcher), laura.imaoka@utdallas.edu (L.B. Imaoka), david.osullivan@auckland.ac.nz (D. O'Sullivan).

https://doi.org/10.1016/j.diggeo.2024.100101
Received 23 August 2023; Received in revised form 23 September 2024; Accepted 2 October 2024

relations under which they are developed and distributed. Recognizing that software undergirds the vast array of digital technologies in urban life and that code is itself politicized (Coleman, 2009; Couture, 2019; Pope, 2016; Tveten, 2016), this article uses the examples of the design, development, and adaptation of the GeoJSON and the Esri Shapefile data interchange formats to reexamine how dead labor, and power, is solidified and distributed across the ecosystems of for profit technology firms. Core to our argument are the concepts of dead labor and the mode of production in the context of software production and use. Dead labor refers to labor that is crystallized or ossified in tools or machinery (Marx, 1992; Wendling, 2009), which in the case of software, includes programs, scripts, standards, and other computing technologies. Like traditional tools and machinery, software is material, but differs in that software is not necessarily tangible. The mode of production refers to the social organization of production, including both the physical means of production and the social relations that define how production occurs (Marx, 1992; Wendling, 2009). Through the lens of the mode of production—power and social relations, software development techniques, and technologies that these file types are part of—the article highlights the shifting terrain on which digital labor is conducted, expropriated, and reified. GeoJSON and Shapefile were both created as open-source standards, but have both become integral to the functioning of closed-source, private products: GeoJSON forming the backbone of many Mapbox products and variants of the Shapefile undergirding some of Esri's python programming environment.

The article proceeds as follows: First, we outline the history of Free and Open Source Software (FOSS) to highlight the emancipatory ideals and contradictory visions of how software is produced, traded, distributed and used. Second, the two specific cases—of the company Mapbox and data format GeoJSON, and of the company Esri and the Shapefile—are examined in detail. Although these two cases appear as partial inverses, they both highlight how Free and Open Source Software (FOSS) operates as an integral component of for-profit software ecosystems that appropriate and expropriate dead labor. These cases develop an idea of not Free and Open Source as an alternative to for profit modes of production and distribution of code and ideas, but of *Free and Open Source and For Profit* as the dominant paradigm through which the production of digital technologies remains encoded in larger systems of capital accumulation.

2. Histories of FOSS

The history of free and open-source software (FOSS) is inextricably tied to the capitalist domination of digital markets, offering a temporary reprieve from monopolistic control while simultaneously being shaped by those very dynamics. These dynamics mirror more general software markets, such as office productivity software—a market still dominated by Microsoft and Google's productivity suites (Cusumano, 2019)—graphic design—where Adobe's Photoshop and Illustrator dominate (Bansal et al., 2022)—or commercial game engine software—where Unity for mobile and Unreal Engine for large-studio games are beginning to dominate the industry (Toftedahl & Engström, 2019). Viewed from the vantage point of various kinds of geographical computing, the tension between software laborers' recognition by proprietary software companies and the desire for software laborers to share (infrastructural) code sets a foundation for the divergent ecosystems of spatial file formats. This tension exemplifies how the productive forces behind FOSS labor practices, development practices, and corporate competition reveal the inherent contradictions of capital in technological development.

As Kelty (2008) illustrates, FOSS emerges from the contradictions of capitalist property relations, where collective labor is harnessed but remains subjugated to the legal mechanisms of control, such as software licenses, open standards, and code-sharing arrangements. These dimensions of FOSS each have their own complex history intertwined with the others. We loosely follow Kelty's comprehensive account without highlighting specific FOSS practices on which each historical episode bears most directly. Kelty draws attention to the complexity of software, but does not examine how the labor of programming has been persistently undervalued, an aspect we want to highlight.

The origins of *programmable* computing reflect the militarized and state-driven appropriation of labor under capitalism, where workers' contributions are obscured by a focus on the technological artifact. The first such computer was the bespoke Electronic Numerical Integrator and Calculator (ENIAC) computer developed during World War II to calculate ballistic missile trajectories. Programming ENIAC in its early years involved configuring switches and rewiring panels by workers collectively known as the "ENIAC girls" (Kennedy Jr, 1946). With the official job title of "computer," these women were responsible for running and operating ENIAC (Abbate, 2012; Fritz, 1996). Laboratory managers so misunderstood the complexity of programming that the women programmers had to create their own social structures, labor processes, and divisions of tasks together, socially, on the shop floor (Hicks, 2017; Light, 1999).

The fetishization of the machine over the labor behind it exemplifies Marx's concept of commodity fetishism, where the valorization of technology erases the social relations and labor necessary for its production. Although the true innovation of ENIAC lay in its programming—the labor required to make the machine functional—popular and scientific attention fixated on the physical machine itself (Hartree, 1946; Kennedy Jr, 1946). This focus continued to obscure the labor and social structures essential for the laborers to function, reducing the visibility of the workers responsible for creating and maintaining the technology. Even as later modifications enabled ENIAC to run stored programs, the labor of programming remained intense and vital, further demonstrating that technological advancement did little to diminish the exploitation of labor (Bullynck & De Mol, 2010).

The early commodification of computing in the 1950s transferred the labor challenges of the military-industrial complex to the corporate and governmental sectors, as capital sought to exploit the productive capacities of computing, agnostic of the challenges of programming labor. Early Univac and IBM mainframe computers shipped with little to no operating software. Without installed code, there was significant duplication of effort for basic functions between purchasers of early machines; and, as a result, the first computer user groups emerged fostered initially by vendors—but driven by users and their needs. The most well-known were the aptly named SHARE[1] (usually capitalized, but not an acronym), for the IBM 704, and the later GUIDE for the IBM 702, 705 and 650 series (Akera, 2001; Campbell-Kelly, 2004). These groups developed and shared extensive libraries of utility programs that performed basic tasks such as mathematical functions and assemblers (to make development of machine code more human-friendly). User groups were, "a conduit through which users' needs were channeled into the manufacturers' programming departments. It was through the efforts of user groups that consensus was reached on the types of software that users needed" (Campbell-Kelly, 2004, p. 54).

While early practices of code sharing and distribution may seem to suggest that software is inherently a collective product, the reality is far more complex. However, the reality is more complicated. These forms of production were subsumed under capitalist property relations, fragmenting what could have been a shared digital commons, and subject to competing computing architectures. Early user groups like SHARE were essential for addressing the redundancy of programming work, but these groups were bound by the specific architectures of the machines they used. Code that ran on an IBM machine, for example, was incompatible with other machines, creating isolated communities of users. It wasn't until the widespread adoption of high-level programming languages like

---

[1] SHARE was preceded by the Project for the Advancement of Computing Techniques (PACT).

FORTRAN and COBOL in the 1960s that more general code-sharing became feasible. Even then, compilers had to be tailored to specific machine architectures, limiting true interoperability (Campbell-Kelly, 2004, pp. 34–36). This technological fragmentation, coupled with the proprietary interests of corporations, reveals the contradictions in the commodification of software labor, as the free exchange of code became increasingly difficult in an industry driven by capital accumulation.

Although code and processes were shared among ENIAC's first coders out of necessity, today's ethic of sharing software code did not enter mainstream until the development of Unix in the 1970s (Ritchie, 1984; see also accounts in Kelty, 2008 Chapter 4; Haigh & Ceruzzi, 2021, pp. 32–25; Weber, 2004, p. Chapter 2). Unix's success, driven by its architecture-neutral design, facilitated the abstraction of labor from specific machine dependencies. In fact, the C programming language (Ritchie et al., 1978) was written in tandem with Unix to facilitate the development of Unix (Ritchie, 1984). Because it was written in C, and not in an architecture specific machine code, porting Unix to other platforms required that only a C compiler be available—greatly smoothing widespread adoption. Arguably the definitive instance of the genre (Kelty, 2008, pp. 128–131), Unix is both an operating system design and a collection of small programs easily chained together to perform complex tasks. This structure makes it ideal for extension and further evolution by others, for example the addition of another small program to handle a common task.

By the 1980s, hardware standardization and the competition between Unix-based workstations and the IBM Personal Computer (PC) highlighted the role of capital in shaping the infrastructure of computing, its labor relations, and the emerging corporate desktop market. By this time AT&T had been deregulated and was pursuing a more proprietary approach to Unix—but the proliferation of incompatible Unix variants during the earlier period of benign neglect, along with rapid advances in the Intel chips on which the much cheaper IBM PC was based proved fatal, and the Microsoft-Intel platform became dominant.

Despite Unix's commercial decline in the 1980s, its entrenchment within university research centers underscores how academic labor and experimentation have historically contributed to the reproduction of capitalist technologies. It was in this setting that it indirectly contributed to the emergence of the novel software licenses that are a key ingredient of FOSS. The hacker ethos of the university computer lab, rooted in collaborative code-sharing, inevitably clashed with the capitalist imperatives of commercial software production, exposing the tensions between collective labor and private accumulation. When a port to Unix appeared in 1981, its author James Gosling, tired of maintaining it, decided that its future was best guaranteed by selling the rights to UniPress. This sparked an extended controversy over the legalities and ethics of the ownership of program code (Kelty, 2008, pp. 183–199). This also coincided with the extension of copyright protections in the United States to program code in 1980, and spurred Richard Stallman, a longtime denizen of the MIT computer lab (Levy, 2010)[2] into announcing plans to develop a free (to share and modify) Unix from scratch under the moniker GNU (a recursive acronym for GNU's Not Unix). Stallman eventually pragmatically (if unhappily) accepted that the only way to protect his (later the Free Software Foundation's) right to share GNU programs was through a novel copyleft license. Stallman introduced the GNU General Public License (GPL), which permits free copying, modification, and distribution of software *subject to the same license being applied to any derived products*. The GPL is a recognition that for program code to be a shared public good under capitalist systems of property rights that status and the accompanying use-rights must be legally asserted through copyright (Raymond, 1998).

Ironically, capitalist efforts to police intellectual property catalyzed the FOSS movement, as programmers sought to reclaim control over their labor and the products of their work. For example, in the late 1980s AT&T barred the study of Unix source code in university classrooms, prompting the development of Minix by Andrew Tanenbaum (1987). Minix is a stripped down Unix-like operating system, distributed with his textbook and installable on cheap PC hardware, and became the starting point for the development, only a few years later, of Linux. This project, developed from early in its evolution on open-source principles, and released under the GPL, is called the first truly open-source project by some (Raymond, 1999a). The Linux kernel has since come to dominate many spaces outside the desktop environment, such as servers, the cloud, high performance, computing, and embedded systems.[3]

The GNU General Public License (GPL), often framed as a subversion of capitalist copyright law (Williams & Stallman, 2010), exposes the contradictions between the freedom to share software and the imperatives of capital accumulation. While the GPL permits free copying, modification, and distribution of software, it enforces the requirement that any derivative works be distributed under the same license, thereby creating a form of legal protection for software as a shared public good. However, despite this legal framework aimed at fostering collaborative development, the GPL has not been widely embraced by businesses (Lea, 2000), which compels the sharing of all subsequent modifications. Linux is distributed under the GPL but many other open-source projects adopt licenses approved by the Open Software Initiative (OSI). OSI was established in 1998 with the express purpose of providing a business-friendly way for corporate actors to open source code while protecting their commercial interests, in the immediate context of Netscape's decision to open the program code of the Navigator browser (see Raymond, 1999b; Weber, 2004; Kelty, 2008, pp. 99–112).

As Kelty (2008) argues, free and/or open-source software is an ongoing development process—thus part of a broader *production* process, constantly shaped by the dialectics of capital and labor. FOSS is not an inherent attribute of the end product as it is a nearly immaterial and intangible collection of digital ones and zeros without regard to its legal status. Throughout the foregoing we have noted the complex decentralized coordination of labor involved in the production of software. The mutual support of early computer user groups, the distributed proliferation of modifications to Unix, the gendered division of labor, and the devaluing of women's work (which necessitated the bottom-up organization of the ENIAC girls) have all been instrumental in shaping the social norms around software use and development.

The emancipatory potential of labor opportunities and skill-building within the FOSS community must be critically examined in the context of entrenched power relations and hierarchies, which consistently undermine these possibilities. While the decentralized and open nature of FOSS initiatives purports to encourage democratic participation and foster skill development, these opportunities remain unequally distributed. The gendered division of labor—evident since the early days of computing with the devaluation of the ENIAC girls' contributions—demonstrates how social norms and institutional hierarchies continue to perpetuate inequalities even within ostensibly open and collaborative environments. This contradiction reveals a deeper issue: although FOSS projects offer platforms for skill enhancement and professional growth, they simultaneously replicate and often exacerbate existing social and economic disparities. The power dynamics within these communities ensure that the benefits of open-source development disproportionately accrue to those already privileged by existing structures—especially large corporate entities—thereby fundamentally limiting the radical, emancipatory potential of such engagements. Thus,

---

[2] Levy's account is highly readable, but as Kelty archly describes it “edenic” (Kelty, 2008, page 332). Reader beware!

[3] Unix-like operating system kernels are also the basis of Apple's iOS and Google's Android mobile platforms while Apple's desktop MacOS is Unix-derived BSD system. There is now a Linux compatibility layer, the Windows Subsystem for Linux, on MIcrosoft's Windows, perhaps the ultimate recognition of the platform's significance.

while FOSS appears to promote a labor model that empowers individuals through networking and collective skill-building, it must also be scrutinized for how it reproduces the very power structures it claims to transcend.

The persistence of these hierarchies of power and economic disparity—once embodied by the experiences of the ENIAC girls—continues to shape contemporary software production. Platforms like GitHub, for instance, exemplify the integration of code sharing and mutual support within the production process, yet also reflect deeper structural inequalities. Although GitHub fosters collective problem-solving through features like issue tracking, discussions, and pull requests, this communal approach has now become the standard model for labor organization in modern open-source communities. However, the gendered division of labor remains a constant. D'Ignazio and Klein (2020) have shown how gender and power intersect within GitHub, noting that contributions from women are accepted at significantly lower rates than those from men. Similarly, Stephens (2013) demonstrates that asymmetries in gender representation persist in platforms like Google Maps and OpenStreetMap. Mahmoudi (2017) further documented how women in software labor have had to form informal groups to combat the devaluation of their work, with the hiring process reflecting these entrenched biases. As Lunn and Ross (2021) have noted, women often occupy roles as recruiters and hiring managers, yet technical interviews and decision-making power remain dominated by men (Mahmoudi, 2017). The persistence of such structures in FOSS communities reflects the limits of its emancipatory claims.

Before turning to the specific means by which dead labor is captured and extracted from said communities in the geospatial sector of today, four points are worth emphasizing. First, the ascendancy of FOSS has remained largely within *infrastructural* code. User-facing programs, such as ArcGIS Pro or Microsoft Office, remain largely proprietary and closed; however, they are built upon and leverage networks of code written and developed with open-source licenses; as the subsequent sections reveal this is a key explanation for for-profit corporations supporting FOSS development.[4] Second, historically, many of these practices developed in environments shielded from market forces—such as government labs and academic departments, sometimes funded by the state. Such spaces, where funding and prestige accrue through avenues other than pure capital accumulation, hide the very real exploitation and appropriation of dead labor in software. This in turn has contributed to myths and rhetoric around creativity, sharing, and improvisation (clever hacks) that remain central to the attractions of openness in code (Levy, 2010; Raymond, 1999b). Third, we choose to examine GeoJSON and the Esri Shapefile as examples of different technology stacks and organization profiles. That is, as cases, they are archetypes of open source and proprietary development. Yet, as we show, this distinction becomes complicated when viewed from the perspective of the capitalist mode of production. Our focus on file formats should be read as a dismissal of other geospatial technologies such as the GIS software program QGIS, the programming language python, or the GeoPackage file format. Finally, the iterative, cumulative, and historical nature of this development should not be forgotten. End-users of software interact directly only with the final layers of code, not realizing (nor likely very much caring about) the decades of development on which it rests. In the following sections using two examples from major players within the geospatial industry, we show how these underlying points and historical contexts have allowed for the emergence of an idealistic ethos of FOSS that elides the ongoing dead labor produced for (and often by) for profit endeavors.

3. GeoJSON and MapBox

GeoJSON, a non-acronym portmanteau of "Geo" and JSON,[5] is an open standard interchange file format for storing, exchanging and distributing geospatial data and files originally developed by Howard Butler of Hobu Incorporated, Martin Daly of Cadcorp, Allan Doyle of MIT, Sean Gillies of UNC-Chapel Hill, Tim Schaub of OpenGeo, and Christopher Schmidt of MetaCarta between 2006 and 2008 (GeoJSON, 2008). GeoJSON is a spatial file format that was designed to work well for both web-based and desktop geographic information system (GIS) software applications. As the previous section demonstrated, while there is no universal agreement on the meanings of "open" or "open standards," the GeoJSON specification underwent a public drafting process and retains a Creative Commons Attribution 3.0 license allowing others to copy, redistribute, transform, build upon "for any purpose, even commercially" (Creative Commons, 2007); this distinguishes it from less permissive copyleft licenses which make derivative commercial use extremely difficult.

GeoJSON rapidly evolved from around 2007 due to the efforts of open-source developers attempting to create an open file format as an alternative to existing, more limited formats. In particular, development for the web was a primary motive and other formats either had significant web-based shortcomings, were proprietary, or were combinations thereof (lists.geojson.org, 2007). For example, Esri's Shapefile format was both proprietary and of limited use in web environments To maximize adoption, this new open file format had to leverage a popular and existing information architecture. During development, significant debate occurred over the use of the more well established eXtensible Markup Language (XML) or the newer JavaScript Object Notation (JSON) information architectures. By the time an email list-serv and a group wiki were created to house and document discussions over the development of a new geospatial file format, JSON had all but been decided on because of its widespread appeal, its smaller footprint, and the speed that it could be read by existing software libraries (lists.geojson.org, 2007). Despite moving forward with JSON, the discussion continued, as captured by this message from an early contributor:

> "I'm sure [that] this list will be rife with flaming regarding whether or not XML is good, evil etc., and the same for JSON. I think those are good discussions to have, but I am going to call out my emails on those topics with <religious-war> tags, and try not to mix them up in other discussions of how to use JSON. Because, regardless of your position, using JSON in one form or another is practically a necessity in dealing with browser based apps" (lists.geojson.org, 2007).

An extended argument over XML and JSON ensued, tagged appropriately as a "religious-war." Notably, these discussions continued long after the listserv had been formed and named the Geo*JSON* listserv.

Using JSON as the base information architecture provided many real advantages for GeoJSON's early conceptualization and subsequent adoption in part due to JSON's widespread use in web technologies and open-source projects alike. Web interpreters could quickly (and asynchronously) parse and retrieve JSON data. Originally created for JavaScript (or JS, hence its name), JSON had already become widespread in open-source languages like PHP, python, perl, and ruby before GeoJSON was developed. Further, JSON data is relatively compact. On the web, the file size of GeoJSON was smaller than competing structuring formats like XML or GeoRSS. This was debated on the email listserv, but it is now generally agreed that JSON is more compact than XML.[6] By using JSON, the early contributors could easily express vector geographic data in a structured format, relying on the work already completed in the JSON project (lists.geojson.org, 2007).

---

[4] While we focus on the mode of production in this paper, we are excited by the publication of "The Beyond ESRI Resource Guide" featuring a number of mainly open source alternatives to Esri's portfolio of mapping products (Bosse et al., 2024).

[5] JSON itself is an acronym standing for JavaScript Object Notation.

[6] GeoJSON only requires an opening tag while XML and GeoRSS require an opening tag and a closing tag.

While the stated goal for the group was to create an open standard geospatial file format, the speed at which the file format went from conception to adoption was relatively fast. The listserv formed in March of 2007, the first request for comment on the GeoJSON structure was up by June, and by September numerous companies and software projects had completed support for GeoJSON or stated their intentions to do so. This was not lost on those observing the fledgling format, as observed in *Directions Magazine*:

> "Although this standard is still developing, it is relevant to know that GeoJSON has already been adapted for the open-source project GeoServer and it has been proposed for inclusion in MapGuide open-source. ... Essentially, GeoJSON was adapted from JSON and then became geospatial mainstream technology in less than a year" (Andrews, 2007).

The rapid ascent of GeoJSON from design spec to geospatial mainstream technology is best understood within the temporal context of its occurrence. In October 2004, Google acquired two companies whose technologies they sought to combine. First, Google acquired an Australian web mapping company, Where 2 Technologies. Google also acquired Keyhole, Inc. which was originally funded by Sony, NVIDIA, the CIA's In-Q-Tel, and other private investors. Keyhole's main product was Earth Viewer, which eventually became Google Earth and released it in that same year (Lee, 2010). By integrating technology from Where 2 Technologies, Google was able to develop Google Earth into a "slippy" web interface. In this context, slippy refers to maps that enable users to click and drag the map, with content dynamically loading around the area of focus in real time. Google effectively ported the Google Earth desktop interface to the web, added additional points-of-interest and road networks, and debuted Google Maps in 2005. Additionally, Google released access to points-of-interest and wayfinding through proprietary, but publicly usable, Application Programming Interface (API) in June 2005 which allowed developers (Crampton, 2008; Lee, 2010). The slippy Google Maps browser interface instantly antiquated alternative mapping products from competitors Microsoft, Yahoo and Mapquest (among others). To add additional data layers to their slippy maps, Google used Keyhole Markup Language (KML), an open, but cumbersome, XML based format developed by Keyhole for their Keyhole Earth Viewer.

With its dominant market power, Google began to develop and expand a foothold in geospatial technologies throughout the mid-2000s. Existing open-source geospatial software, such as Geographic Resources Analysis Support System (GRASS) and GeoTools, presented powerful GIS software packages for the *desktop*. Google Maps' integrated roads, points-of-interest, and satellite imagery accessible via slippy *web* maps gained immediate popularity. "[T]he pan and zoom of 'slippy maps' have become an everyday part of life" (Crampton, 2009, p. 92). As Google entered this space, the notable absence of successful and widely adopted open-source geospatial formats, standards, and software prompted the O'Reilly sponsored Where 2.0 conference in 2005.

> "Location-aware technologies like GPS, RFID, WLAN, cellular networks and networked sensors are enabling an ever-growing array of capabilities, from local search, mapping, and business analytics to enterprise integration, commercial applications, and software infrastructure. The first O'Reilly Where 2.0 Conference has been created to explore the emerging consumer and enterprise ecosystems around location technologies—ecosystems that can radically change the way we work and play. Where 2.0 will take place June 29-30, 2005 at the Westin St. Francis in San Francisco, California" (O'Reilly Media Incorporated, 2005).

Following this conference, MetaCarta, which had already received significant backing from the US Government and US Department of Defense contractors, created an open-source product Open Layers. MetaCarta debuted a mapping interface that had a similar technology stack to Google's and used a file format nearly identical. At the same time, the open-source Geospatial Foundation (OSGEO, which runs the FOSS for Geospatial or FOSS4G conference series) formed to "support and promote the collaborative development of open geospatial technologies and data" (Open Source Geospatial Foundation, 2006). By the end of 2007, Open Layers had become an official project of OSGEO and had begun to incorporate the promising GeoJSON open standard.

GeoJSON development then directly coincided with the growth of open-source geospatial software and contributed to its success and adoption. Over the next several years, project and funding collaborations from organizations like Development Seed, The John S. and James L. Knight Foundation, Foundry Group, OSGEO, and Open Layers supported significant development on the web-based open-source geospatial technology stack and the diffusion of GeoJSON as the de facto geospatial data format for web applications. It was in this context that MapBox emerged from Washington, D.C. based analytics firm Development Seed. By 2010, developers at Development Seed had gathered extensive mapping expertise and had contributed to the development of GeoJSON both through funding and technical contributions. MapBox, building from its early successes with open source Leaflet Javascript Library, was spun out to offer map customizations to other non-profit organizations that often leveraged their expertise in open-source technology stacks. While today GeoJSON's interoperability in open-source geospatial software is a de facto norm, its diffusion is neither entirely due to nor limited to FOSS nonprofits. The open standard format of GeoJSON was crucial in supporting a new cadre of firms in the startup economy and, in turn, its widespread adoption was furthered by these firms. The relationship between GeoJSON and the burgeoning start-up MapBox, estimated as worth approximately $700 million in 2017 (Carson, 2018), serves to illustrate how FOSS intersects with, supports, and is supported by profit driven technology firms.

Mapbox was funded by Development Seed and Knight News Challenge to provide mapping service to non-profits and improve the core infrastructure behind OpenStreetMaps (MacWright, 2012). Despite originally being launched as a FOSS company intended to serve non-profits, it has since expanded to provide commercial mapping for companies like *The New York Times*, *REI*, *Snap Inc., and CNN*. These changes occurred over time as Mapbox slowly incorporated individuals and technologies into their corporate and technological stack—these include developers that worked on the original GeoJSON specification and technologies like spatial analysis library Turf.js (Bratton and Author, 2016). As Mapbox has continued to grow from spin-off start-up meant to service non-profits with spatial analysis and visualization towards for-profit enterprise built off of FOSS software, so too have the tools they provided changed.

The core pillar of interactive visualizations provided by MapBox services is Mapbox GL JS, described as "a JavaScript library for interactive, customizable vector maps on the Web." (Mapbox, 2023). An open-source library since its inception, Mapbox GL JS changed its license with the release of version 2.0.0 in December of 2020:

- "mapbox-gl-js is no longer under the 3-Clause BSD license. By upgrading to this release, you are agreeing to Mapbox terms of service. Refer to LICENSE.txt for the new licensing terms and details. For questions, contact our team at https://support.mapbox.com" (Mapbox, 2020).
- "Beginning with v2.0.0, a billable map load occurs whenever a Map object is initialized. Before updating an existing implementation from v1.x.x to v2.x.x, please review the pricing documentation to estimate expected costs" (Mapbox, 2020).

Responses to the change were swift and varied according to the interests of the authors. For example, Mapbox competitor Carto's founder used it as an opportunity to tout their technology and commitment to open-source ideals (de la Torre, 2020). The front page of ycombinator's Hacker News site, a popular site for developers, featured a post by Joe Morrison, a developer for geospatial technology and research firm Azavea, describing the "death" of an open business model as firms, "[e] ventually, if they're successful, they will be forced to choose between betraying their loyal early adopters and dying a long, slow death" (Morrison, 2020). In this telling, Mapbox was simply acting as "rational economic actors" (Morrison, 2020) should, or as long-time employee Samuel Bemel Benrud tweeted, "Mapbox is trying to become sustainable, finally" (Mapbox, 2020).

A cynical response was given by Paul Ramsey who sees Mapbox not as a software company, but as "always [having] been a data and services company" (Ramsey, 2020) Ramsey also notes that this is not the first time Mapbox has shifted one of their offerings as TileMill, an open-source tool, became Mapbox Studio, a closed one (Ramsey, 2020). Both Ramsey and Morrison describe what they see as the flaws of open core business models—i.e. business models in which the core technology is open-source, but the firm sells the expertise to put the various open-source technologies together and manage them in a (relatively) easy to use stack. The problem with this model, both contend, occurs when larger firms simply copy existing open-source code into their products, effectively nullifying the advantages offered by Mapbox's enterprise solutions. In this case, this occurred when Microsoft's Azure Maps openly, and legally, incorporated Mapbox GL JS into their services, which was publicly celebrated by Mapbox at the time (Lee, 2019).

Ramsey draws attention to similar approaches, and outcomes, taken by other start up firms. In particular, a quote by Dev Ittycheria, the CEO of MongoDB, a cloud-oriented database program, suggests that open-source is always a means to an end: "We didn't open-source it to get help from the community, to make the product better. We open-sourced as a freemium strategy; to drive adoption" (Ramsey, 2020). Regardless of the intentions behind Mapbox's adoption, incorporation, and development of open-source technologies, like GeoJSON, into their stack, a specific sequence emerges. First, open-source technologies are incorporated and developed to drive adoption; during this phase existing disruption occurs by "undercutting costs via VC [venture capitalist] subsidies" (Benrud, 2020). The congealed, dead labor that creates and maintains the open-source technology stack is expropriated into a for-profit model that sells expertise and ease of use of said technology.

Open-core becomes incorporated into a still larger platform. For Mapbox this occurred when Microsoft's Azure Maps cloud-service openly, and legally, incorporated their core visualization library into their platform. At that point, venture capital backing can no longer offset costs based on promises of market disruption and future growth as a much larger firm is able to leverage their infrastructure and market dominance against the smaller start up. The start up must adapt and, as Mapbox and MongoDB demonstrate, they often do so by abandoning their previous open-source commitments to one degree or another.

In this section, we have charted the history of GeoJSON and how its incorporation and development have been tied to the history of MapBox. In particular, we've shown a process by which labor becomes dead labor embodied in open-source technologies, and then leveraged into for-profit endeavors. Where this section focused on moving from an open technology through to a for-profit firm, the next section expands on how existing for-profit firms integrate with, manipulate, and profit from open-source endeavors; in this case, through an examination of how the corporation Esri has supported, adopted, and influenced open-source geospatial standards.

4. Esri and shapefiles

Esri, short for Environmental Systems Research Institute, Inc., considers itself the "global market leader in GIS" (Esri, 2023). Since its founding in 1969, Esri, based on a $5000 loan from the founder's mother (Esri, 2019), the Redlands, California-based privately-held company has enjoyed significant growth for more than five decades. Esri's 2017 revenue is estimated at over 1.1 billion USD (Mihindukulasuriya, 2017). Its proprietary software and services are used in diverse sectors such as scientific research, military and defense, environmental protection, community resilience following large-scale disasters, police use for crime prevention, K-12 education, and commercial business. Esri has developed a near monopoly in government with many national governments, 50 US states, and 20,000 cities employing its software and services (Esri, 2023b). This is complemented by similar use in the commercial realm with 50 % of Fortune 500 companies paying Esri subscriptions (Esri, 2023b). Esri has a particular dominance in the local, state, and federal government sectors and has built a strong pipeline into its products through a variety of outreach and partnership efforts with Thatcher et al., 2016; Imaoka, 2021). Together, they constitute an estimated 45 % of the world's GIS market, by far the largest for a single firm (Business Wire, 2019).

Despite an explicitly for profit, market driven self-narrative, Esri considers itself an advocate for open-source geospatial standards. This is often articulated through a lens of inclusivity and brand management (Imaoka, 2021). As Esri IT Strategies Architect Victoria Kououmjian explained in the 2012 edition of ArcUser:

> We're all in this together. If you want to further the cause and broaden the application of GIS, it's time to bridge the gap that apparently still exists in the geospatial community between open and closed source and transcend the bad game of tug-of-war that has been going on between and within these GIS partitions. The interesting thing about tug-of-war is that—unless the rope breaks—both sides are exhausted in the end (2012, p. 34).

Towards this end, the company is a highest-tier sponsor of OpenStreetMap's State of the Map conference alongside Mapbox and others (OpenStreetMap, 2021). Esri sends representatives to speak there and at other open-source oriented conferences such as FOSS4G. Esri also releases its own open-source projects such as the ArcGIS Editor for OpenStreetMap, which is licensed under the permissive Apache 2.0 license (meaning the code can be used in projects with other licenses, including for profit ones) and are gold-level sponsors of the Geospatial Data Abstraction Library Project (GDAL, 2023). As early as 2012, Esri was proposing a middle in which their most prominent, proprietary software offering, ArcGIS, was able to interface with and leverage over 100 open-source-projects, APIs, and libraries; a number which has only grown in the years since (Kouyoumjian, 2012). While framed in terms of improving interoperability, the history of Esri's development and, partial, open-sourcing of their shapefile data format alongside the embracing of the alternative GeoPackages format presents a more nuanced history.

The Open Source Geospatial Foundation (OGC) originated in 1994 as an international non-profit institutional arrangement set to build a more open and transparent context for standards development and adoption. Its certification program is meant to ensure that compliant products can be integrated as solutions regardless of their vendor. While testing a product's compliance is free, there is an annual fee for trademark licensing to be marked as "Certified OGC Compliant" (Open Source Geospatial Foundation, 2006). The OGC's stated commitment is to improve access to the world's geospatial data. More specifically, to make location information "FAIR – Findable, Accessible, Interoperable, and Reusable," which compliments the popular suppositions of the FOSS community.

For Esri, this advocacy for open access to geodata can be found in their "open system strategy" (Kachelriess, 2012) which continues as an important plank of the brand's strategy. Esri's continued support of OGC's endeavor encompasses adopting formats outside their own. Despite having internally developed the Shapefile, Esri supports ArcGIS's use of OGC's compliant file format, GeoPackage. The company was an early adopter of GeoPackage, even supporting it before it was approved for OGC membership in 2014 (Sankaran, 2014). GeoPackage is an open, platform-independent, standards-based universal file format for geodata that includes vectors, raster maps, extensions, and matrix sets. Besides being open and OGC standard, the SQLite-derived container gained praise for having a broader implementation, being lightweight but as fast as geodatabases, and having easier file management than Esri's Shapefile.

The history of the shapefile and its emergence as the de facto standard for spatial data for over a decade illustrates an inverse process to that which MapBox undertook as described in the previous section. Whereas, when a larger competitor began integrating MapBox' core product into their cloud system, they responded by closing off and making proprietary some of their stack, Esri used ostensibly open technology to assert market dominance through interoperability. The Library of Congress' entry for Esri's Shapefile states that it was published in 1998 and describes it as "[a]lthough proprietary, the intention behind publishing the format was to encourage its use for interoperability among geographic information system (GIS) applications" and lists it as a "preferred format for GIS vector data" (Library of Congress, 2020). This follows Esri's currently available "Esri Shapefile Technical Description Document" which has a copyright date for 1997 and 1998 and aligns with the OGC certification of several Spatial Database Engines in 1999 (Esri, 1998; Open Source Geospatial Foundation, 2006). However, it does not fully correspond to the historical record, and the discrepancies provide important insights into how and why Esri adopted a data standard that is simultaneously open and proprietary.

The shapefile format was created alongside ArcView 2.0 which was released in 1993. Further, the shapefile format—and how to translate into and out of it—were a topic of active discussion on the usenet board comp.infosystems.gis from at least 1995 (Hammond, 1995). Referring to a 1995 "ARCVIEW Shapefile Technical Description found in the ARC/INFO White Paper Series," a poster asks a series of technical questions pertaining to the placement of nodes in relation to polylines and arc segments with the intent to "importing ARC Shapefiles into a groundwater modeling program" using the programming language C (Hammond, 1995). Other early discussions similarly revolve around importing shapefiles between programs. In a thread from September of 1996, a user asks for "scripts to convert ArcView shapefiles to MapInfo files" and receives responses ranging from free scripts available for download to multiple for-sale programs that will handle "ALOT [*sic*] more than just Shape to MIF translation" (Aldridge, 1996).

These early discussions are instructive both as to the historical lineage of Shapefiles as well as the early and ongoing interest in interoperability of spatial information. On one level, they help explain the 1998 whitepaper that fully documents shapefiles as well as Esri's ongoing emphasis on incorporating interoperability into their business structure. Whereas, when Microsoft incorporated part of Mapbox' technology into their system, Mapbox responded by making updates to said technology proprietary. As the more dominant player in the market space, Esri on the other hand used the technical documentation of shapefiles in an open whitepaper as a means to preclude smaller organizations from establishing themselves within their market. With shapefile documentation fully open, there is no longer a space for $499 3rd-party software that will convert between shapefiles and other formats (Aldridge, 1996). Esri used their market position to drown out competition *through* embracing open ideologies and practices.

The embrace of FOSS and OGC standards increases the interoperability and use of Esri's products. Similarly, developing open-source data formats and scripts helps maintain Esri's base of users, keeping them from needing to work fully within FOSS communities. An example is ArcGIS for OpenStreetMap (OSM), an open-source add-on for ArcGIS Desktop. Esri's users can download data from the OSM server to a local geodatabase. Users can then use ArcGIS to edit the data. The editor maintains OSM geodatabase's scheme and symbology and allows users to upload their changes back into OSM's servers, all without the end-user ever directly accessing OSM's interface. Esri's users can thus participate in the OSM community without leaving the confounds of proprietary software. APIs and Software Developer Kits released by Esri also work within open-source development environments, holding potential to extend into new applications that Esri can later appropriate into their platform.

5. Dead labor and the mode of production

Through the lens of the mode of production, the story of the GeoJSON and the Shapefile situate data standards within a broader set of social relations in the production of geospatial software, products, and knowledge. This approach provides clarity into the production of digital technologies by disentangling the various elements involved. It separates the aspects of labor, development techniques, and digital machinery (together, the means of production) from the aspects of organizations, companies, and licenses (together, the social and technical relations). In doing so, we highlight the ever-evolving terrain on which digital labor (specifically, software labor, as discussed by Dorschel, 2022) is conducted, expropriated, and reified. A key aspect of this analysis is understanding the lineage of both the means of production and the social relations of technology through the accumulated labor of FOSS developers, often referred to as dead labor.

The contribution of FOSS developers—embodied in software and software libraries as dead labor—is far from trivial. In 2011, activist-scholar Marcell Mars estimated that Google search infrastructure had 100,000 web interface servers, 10,000 look queries, and 1000,000 computers housing Google's search index. Each of these servers was based on the software packages of the linux kernel, python, and Apache, representing some of the largest cooperative, free and open-source software packages available (despite that by this time, Google had moved to their own custom linux-based web server with Apache-like software). The linux kernel, between 1991 and 2011 had over 10 million lines of code written. Between 2005 and 2011, 6100 individual developers and over 600 companies contributed to the linux kernel. Mars estimated that Google would have to hire 1000 developers, require $1 billion in non-labor related investment, and would take 12 years for the 1000 developers to recreate the free and open-source software that went into the servers alone (Mars, 2011).[7] This helped propel Google from a handful of employees in 1998 to an initial public offering in 2004 that raised $1.67 billion USD, giving it a capitalization of $23 billion USD. In 2011, when this calculation was made, Google had annual revenue of nearly $40 billion USD and in 2022 had an annual revenue of nearly $280 billion USD. The dead labor of open-source software played key parts of Google's initial growth and its capacity to continue to be profitable.

The importance of dead labor is not limited to companies that are primarily web companies. Apple's early 21st century resurgence can also be traced to how it incorporated and encouraged the production of open-source products into its technology stack; most notably through the Darwin open-source operating system on which macOS and iOS are both based. Darwin was released by Apple in 2000 and at its root is Berkeley Software Distribution (BSD). After Jobs was ousted at Apple, he formed a rival company NeXT and used the permissive BSD-licenses to prioritize the development NeXTSTEP based off BSD. Apple then purchased NeXT in 1997, returning Jobs as CEO, and doing so with a promise that NeXT's operating system would form the basis of Darwin, the open-source foundation of Apple's proprietary macOS operating system. This was, effectively, the birth of macOS and remains the underlying technology that powers every iPhone, iPad, and Macbook (Foresman, 2012).

Esri's dominance within geospatial software is in part attributed to the facade of openness in the definition of the Shapefile as well as how the company leveraged open-source through compatibility. Esri's market dominance means that the dead labor of FOSS developers to create alternative GIS software and formats is captured through the proprietary Esri products which are made to be compatible with other products. Python is a much larger portion of Esri's reformulation of its desktop GIS

---

[7] Mars' estimate is based specifically on the packages: linux-kernel, gcc, openjdk, binutils, boost, glibc, hadoop, python, hue, apache-pig, openssl, apache-httpd, gnutls, hbase, coreutils, zookeeper, sendmail, util-linux, yahoo-oozie, gsasl, sqoop, whirr,and net-tools.

suite of software (from ArcGIS Desktop to ArcGIS Pro), somewhat similar to the way in which Apple used permissive licensing of open-source BSD for its new generation.

On one level, Esri opened their proprietary format to leverage their market dominance and drive out other competitors (those selling translation scripts for shapefiles). On another level, they incorporate open-source tools *within* their existing systems; similar to how Microsoft incorporated part of Mapbox's technology stack within their own Azure ecosystem. In each case, *open-source* licenses do not prevent market manipulation, but rather contribute to it—allowing different entities to drive others out of marketspaces or improve their own technology through the appropriation of labor done within the open-source community. In the final sections, we discuss how these instances suggest a better understanding of FOSS is not as a fully separate alternative to for-profit development of technology (the edenic garden of Kelty, 2008), but rather as part and parcel of such capitalist oriented systems.

6. Conclusion: free, open, and for-profit

In this article, we first traced the historical development of free and open-source software ideologies and licenses. Tracing the development of a self-conscious ‘free’ software movement back through the earliest programmable computers. We emphasized that the rise of FOSS predominantly occurs within infrastructural code rather than user-facing programs. Much of the historical and ongoing development in these areas takes place in environments shielded from direct market forces. In other words, funding and prestige in these contexts come from sources other than, or adjacent to, pure capital accumulation. Additionally, the nature of computation as a “stack” of technologies allows much of the infrastructural work to go unnoticed. For example, users of the latest version of Microsoft Windows seldom realize that the system includes code written in the 1990s.

We then examined two examples that illustrate how the boundaries between open and proprietary code are strategically manipulated for market dominance: the data formats of GeoJSON and Shapefiles. In the first case, Mapbox, initially relying on open standards, eventually closed off its software to secure profit outside of venture capital funding, all while leveraging open-source technologies within its proprietary platform, notably in opposition to Microsoft's integration of its code into Azure Maps. In contrast, Esri's entrenched dominance within the geospatial technology sector allowed it to publicly document its proprietary Shapefile format, effectively eliminating a cottage industry built around file translation, while simultaneously ensuring Shapefiles became the industry's de facto standard. When alternative formats did emerge, Esri absorbed them into its own technology stack, positioning itself as a leader in interoperability and open technology adoption. Our focus on geospatial technologies here is not to suggest that these dynamics are unique to the spatial industry but rather to emphasize their ubiquity across sectors and the critical need for geographers and geospatial scientists to scrutinize these processes. This manipulation of technological standards has profound implications for spatial understanding. For instance, future research could investigate how Esri's rise to dominance, particularly with Shapefiles as the de facto standard, replaced topology-based formats, embedding the spaghetti data model—where spatial interactions and rules are poorly defined—into digital applications, thereby reshaping how space is understood and governed.

The rise of open-standards as a component of technology firms appears to challenge the dominance of corporations like Google and Esri. Using a mode of production lens reveals how, alternatively, the FOSS production models are not contingent, but instead necessary for technology companies because of the free accumulations of dead-labor as free fixed capital. That is, within larger capitalist systems and with the exploitation and alienation of labor, free and open-source *cannot* be understood outside of the for profit imperatives that drive technology firms. Rather, instead of free and open-source (FOSS) these software development workflows and technologies are inevitably best understood as free, open, *and for-profit*. As mentioned earlier, participation in FOSS projects is not solely about contributing to communal resources but also serves as a strategic avenue for developers to network and enhance their own marketable skills. This interaction often opens professional opportunities, as seen in the career trajectory of OSM founder Steve Coast, where contributions to open-source projects can elevate one's visibility and desirability to commercial entities. From the perspective of dead labor, the initial unpaid contributions that developers make to FOSS projects can be viewed as a form of dead labor benefiting the project itself as well as technology firms which use those projects. However, this also enhances the developers' own living labor potential by improving their skills and marketability. It also encourages developers to self-exploit, in competition with other open source contributors, as they aim for future opportunities—fueling a race to the bottom in terms of self-exploitation. Ultimately, the benefits of this labor accrue disproportionately to those already in power or those able to capture that labor.

Data colonialism offers a particularly useful means of understanding this relationship. Following Thatcher et al.'s (2016) formulation of the term, we are here using data colonialism as a metaphor that foregrounds the extractive forces of capital through technology, a concept they draw through Harvey's concept of accumulation by dispossession (Harvey, 2007). This approach is distinct from Couldry and Mejias (2019a) formulation of the term which highlights its non-metaphorical possibility when considered directly with colonialism's relations to extractive capitalism (Couldry & Mejias, 2019b; Mejias & Couldry, 2019); our more limited use of the term highlights the ties between FOSS and profit in three important ways. First, FOSS is inextricably linked, in both name and history, to the very concepts of free and open spaces through which a neoliberal, extractive frontier mythos has embedded itself within tech culture (Manyika et al., 2011; Yonego, 2014). Second, and more centrally, whereas Thatcher et al. (2016) focused on the extraction and commodification of data, FOSS illuminates a situation in which ossified, dead labor is expropriated by profit seeking firms. Third, this formulation creates space to interrogate the role of the state in financing the development of technologies that were later appropriated by for-profit companies into proprietary software and/or FOSS, positioning the state as the handmaiden to dead labor. Examples include the CIA's involvement in what eventually became Google Maps referenced above or the U.S. Army Construction Engineering Research Laboratory's (USACERL) involvement in the original development of GRASS.

Kirsch and Mitchell (2004), following Marx's own work, define dead labor as “work ossified and made concrete in the shape and form of a machine, a building, a finished commodity, a technological artifact, a piece of property, or even nature itself.” They continue:

> “[T]here is social intentionality in turning relationships into things; there are reasons for putting networks together, even if those reasons themselves are highly structured by and determined within the contested relationships that constitute capital*ism* as a social totality.” Kirsch & Mitchell, 2004, p. 696).

FOSS is not exempt from this totality of relations. Emerging, historically, largely within confines that shielded it from direct market forces, its outputs are easily incorporated within larger structures of profit maximization. FOSS, then, promises an alleviation to the falling rate of profit through an altering of creation of dead labor within the organic composition of capital. Whereas in Marx's classic formulation (Marx, 1992), as soon as a competitor is able to produce a similar product at lower cost, a firm must resort to greater exploitation of labor productivity or maintain its advantage through market manipulation (through a monopoly position, opening a new market, etc.); FOSS offers an alternative in which labor costs seemingly disappear (or are lowered) through the transformation of free labor into dead labor usable by circuits of capital. As with other purported fixes to the inherent crises of capitalism, it is crucial to recognize that while FOSS offers a seemingly attractive remedy to labor costs in the context of the falling rate of profit—by shifting towards the production of ossified dead labor—it fundamentally fails to achieve this fully, as sharply illustrated by MapBox's high-profile struggle over unionization. This failure was exemplified by MapBox's widely publicized legal battle with the National Labor Relations Board, culminating in a settlement over unlawful retaliation against organizers (National Labor Relations Board, 2022). Similarly, when Esri encourages its users to contribute to OpenStreetMap through its own software, they are contributing to a FOSS product, while simultaneously keeping users within their ecosystem and developing mapping data (the contributions) which they can draw upon later. When MapBox was threatened by Microsoft's enrollment of their FOSS software into their larger ecosystem, they responded by making future developments proprietary.

There is no intrinsically and inevitably free aspect to FOSS, rather its developments, much like the history of computing itself, reflect particular times and spaces—particular relations—within larger circuits of capitalism. By framing FOSS within these systems of production and their social relations, we are able to call attention to how dead labor—even labor ostensibly given for free—flows through circuits of capital, through networks of computation and exploitation. While outside the scope of this piece, such a framing allows FOSS, and the firms which make their brand identity and profit through it, to be better understood within larger systems of capitalist production. Digital geographers and scholars of technology can and should examine the temporal and spatial relations of currently successful firms that create and leverage FOSS technologies - considering what special relations allow for their ongoing success or if, perhaps, they may find themselves in the position of MapBox at some point.

CRediT authorship contribution statement

Dillon Mahmoudi: Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization. **Jim Thatcher:** Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization. **Laura Beltz Imaoka:** Writing – original draft, Formal analysis, Conceptualization. **David O'Sullivan:** Writing – original draft, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We would like to extend our deepest thanks to the editor, Sophia Maalsen, for her invaluable guidance and support throughout the review process, and to the five anonymous reviewers whose thoughtful and constructive feedback helped chart the course for this paper. Their insights were instrumental in refining our arguments, ensuring that no concept—or codebase—was left unexamined. Any errors, of course, remain entirely our own coordinates.

References

Abbate, J. (2012). *Recoding gender: Women's changing participation in computing*. MIT Press.

Akera, A. (2001). Voluntarism and the fruits of collaboration: The IBM user group, share. *Technology and Culture, 42*, 710–736.

Aldridge, R. (1996). COMP.INFOSYSTEMS.GIS: ArcView shapefile to MapInfo MIF (UsenetArchives.com) [WWW document]. URL https://www.usenetarchives.com/view.php?id=comp.infosystems.gis&mid=PDMyNEJFRTE5LjQxQzZAY2FtYmVyLmNvbT4.

Andrews, C. J. (2007). *Emerging technology: AJAX and GeoJSON [WWW document]*. Directions Magazine. URL http://www.directionsmag.com/entry/emerging-technology-ajax-and-geojson/122845 (accessed 4.19.15).

Ash, J., Kitchin, R., & Leszczynski, A. (2016). Digital turn, digital geographies? *Progress in Human Geography*. https://doi.org/10.1177/0309132516664800, 0309132516664800.

Bansal, J., Balodi, K. C., Jain, R., & Datta, S. (2022). Strategy and business model evolution at adobe: Competing in digital media software industry. *Journal of Information Technology Teaching Cases*. https://doi.org/10.1177/20438869221116391, 204388692211163.

Benrud, B. S. (2020). *Mastodon.Gamedev.Place/@samanbb on twitter [WWW document]*. Twitter. URL https://twitter.com/samanbb/status/1336380594410770434 (accessed 7.29.23).

Bosse, A., Chennault, C., Marimon, P. D., Donnely, F., Estes, L., Heiderscheidt, D., … Spangler, I. (2024). *The beyond Esri resource Guide*. Antipode Online.

Bratton, B. H., & Author. (2016). *The stack: On software and sovereignty*. The MIT Press. https://doi.org/10.7551/mitpress/9780262029575.001.0001

Business Wire. (2019). ARC Advisory Group Report Names Esri GIS Market Leader. https://www.businesswire.com/news/home/20190131005210/en/ARC-Advisory-Group-Report-Names-Esri-GIS-Market-Leader.

Bullynck, M., & De Mol, L. (2010). Setting-up early computer programs: DH Lehmer's ENIAC computation. *Archive for Mathematical Logic, 49*, 123–146.

Campbell-Kelly, M. (2004). *From airline reservations to sonic the hedgehog: A history of the software industry*. Cambridge, Mass: The MIT Press.

Carson, B. (2018). *The right direction: How Mapbox is winning over developers to challenge Google's mapping dominance*. Forbes.

Coleman, G. (2009). Code is speech: Legal tinkering, expertise, and protest among free and open source software developers. *Cultural Anthropology, 24*, 420–454. https://doi.org/10.1111/j.1548-1360.2009.01036.x

Couldry, N., & Mejias, U. A. (2019a). *The costs of connection: How data is colonizing human life and appropriating it for capitalism*. Stanford University Press.

Couldry, N., & Mejias, U. A. (2019b). Data colonialism: Rethinking big Data's relation to the contemporary subject. *Television and New Media, 20*, 336–349. https://doi.org/10.1177/1527476418796632

Couture, S. (2019). *The ambiguous boundaries of computer source code and some of its political consequences. digitalSTS: A Field Guide for Science & Technology Studies* (Vol. 136, p. 21).

Crampton, J. W. (2008). Keyhole, Google Earth, and 3D worlds: An interview with Avi bar-Zeev. *Cartographica, 43*, 85–93. https://doi.org/10.3138/carto.43.2.85

Crampton, J. W. (2009). Cartography: Maps 2.0. *Progress in Human Geography, 33*, 91–100.

Creative Commons. (2007). Attribution 3.0 United States CC BY 3.0 US [WWW document]. URL https://creativecommons.org/licenses/by/3.0/us/.

Cusumano, M. A. (2019). The cloud as an innovation platform for software development. *Communications of the ACM, 62*, 20–22. https://doi.org/10.1145/3357222

D'Ignazio, C., & Klein, L. F. (2020). *Data Feminism*. Cambridge, Massachusetts: The MIT Press.

Dorschel, R. (2022). Reconsidering digital labour: Bringing tech workers into the debate. *New Technology, Work and Employment, 37*, 288–307. https://doi.org/10.1111/ntwe.12225

Esri. (1998). *ESRI shapefile technical description (an ESRI white paper)*. Esri.

Esri. (2019). *Get to know Esri's Jack Dangermond - the man taking maps mainstream [WWW document]*. Esri Singapore. URL https://esrisingapore.com.sg/news/get-know-esris-jack-dangermond-man-taking-maps-mainstream (accessed 7.30.23).

Esri. (2023). *Company | about Esri [WWW document]*. URL: Esri. https://www.esri.com/en-us/about/about-esri/company (accessed 7.30.23).

Esri. (2023b). *Esri fact sheet | media relations press information [WWW document]*. Esri. URL https://www.esri.com/en-us/about/media-relations/fact-sheet (accessed 8.7.23).

Foresman, C. (2012). *The legacy of NeXT lives on in OS X [WWW document]*. Ars Technica. URL https://arstechnica.com/gadgets/2012/12/the-legacy-of-next-lives-on-in-os-x/ (accessed 8.7.23).

Fritz, W. B. (1996). The women of ENIAC. *IEEE Annals of the History of Computing, 18*, 13–28.

Fuchs, C. (2014). Theorising and analysing digital labour: From global value chains to modes of production. *The Political Economy of Communication, 1*.

GDAL. (2023). Sponsors — GDAL documentation [WWW Document]. URL https://gdal.org/sponsors/index.html (accessed 8.7.23).

GeoJSON. (2008). Specification [WWW Document]. URL http://geojson.org/geojson-spec.html (accessed 8.25.16).

Haigh, T., & Ceruzzi, P. E. (2021). *A new history of modern computing*. MIT Press.

Hälterlein, J. (2021). Epistemologies of predictive policing: Mathematical social science, social physics and machine learning. *Big Data & Society, 8*. https://doi.org/10.1177/20539517211003118, 20539517211003120.

Hammond, G. (1995). COMP.INFOSYSTEMS.GIS: ARC shapefile format question for GIS importation (UsenetArchives.com) [WWW document]. URL https://www.usenetarchives.com/view.php?id=comp.infosystems.gis&mid=PDRhbjIxciQxYzdAamFuLmV0LmJ5dS5lZHU%2B.

Hartree, D. R. (1946). The ENIAC, an Electronic calculating machine. *Nature, 157*, 527. https://doi.org/10.1038/157527a0

Harvey, D. (2007). *The limits to capital*. Verso.

Hicks, M. (2017). *Programmed inequality: How Britain discarded women technologists and lost its edge in computing* (1st ed.). Cambridge, Massachusetts London, England: The MIT Press.

Imaoka, L. B. (2021). Branding the geospatial industry. *Communication, Culture and Critique, 14*(1), 109–126. https://doi.org/10.1093/ccc/tcaa025

Kachelriess, R. (2012). DEFINING what is open? *ArcUser, 2012*, 36–37.

Kelty, C. M. (2008). *Two bits-the cultural significance of free software*. Durham and London: Duke University Press.

Kennedy, T. R., Jr. (1946). *Electronic computer flashes answers, may speed engineering: new all-electronic computer and its inventors. 1*. The New York Times.

Kirsch, S., & Mitchell, D. (2004). The nature of things: Dead labor, nonhuman actors, and the persistence of Marxism. *Antipode, 36*, 687–705. https://doi.org/10.1111/j.1467-8330.2004.00443.x

Klein, J. (2021). Cryptocurrency is in a new wild west phase. In *Things are really, really weird [WWW document]*. Vice. URL https://www.vice.com/en/article/88n8dv/cryptocurrency-is-in-a-new-wild-west-phase-things-are-really-really-weird (accessed 6.14.21).

Kouyoumjian, V. (2012). *Open source, closed source: Moving to the middle. 2012* p. 34). Esri.

Lally, N., Kay, K., & Thatcher, J. (2019). Computational parasites and hydropower: A political ecology of bitcoin mining on the Columbia River. *Environment and Planning E: Nature and Space*. https://doi.org/10.1177/2514848619867608, 251484861986760.

Lea, D. (2000). *Concurrent programming in Java: Design principles and patterns*. Addison-Wesley Professional.

Lee, M. (2010). A political economic critique of google maps and google Earth. *Information, Communication & Society, 13*, 909–928. https://doi.org/10.1080/13691180903456520

Lee, T. (2019). *Azure maps adds data-driven styling powered by Mapbox GL [WWW document]*. Medium. URL https://blog.mapbox.com/azure-maps-adds-data-driven-styling-powered-by-mapbox-gl-f14d062b79f9 (accessed 7.29.23).

Levy, S. (2010). *Hackers: Heroes of the computer revolution - 25th anniversary edition, 1* (edition. ed.). Sebastopol, CA: O'Reilly Media.

Library of Congress. (2020). ESRI Shapefile [WWW Document]. URL https://www.loc.gov/preservation/digital/formats/fdd/fdd000280.shtml (accessed 8.7.23).

Light, J. S. (1999). When computers were women. *Technology and Culture, 40*, 455–483.

lists.geojson.org. (2007). *The GeoJSON archives [WWW document]*. The GeoJSON Listserv Archives. URL http://lists.geojson.org/pipermail/geojson-geojson.org/ (accessed 4.16.15).

Lunn, S., & Ross, M. (2021). Ready to work: Evaluating the role of community cultural wealth during the hiring process in computing. In *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. https://ieeexplore.ieee.org/abstract/document/9620686/.

MacWright, T. (2012). Support Development Seed's Knight News Challenge Submission—Development Seed. https://developmentseed.org/blog/2012-03-30-support-development-seeds-knight-news-challenge-submission.

Mahmoudi, D. (2017). *Making software, making regions: Labor market Dualization, segmentation, and feminization in Austin, Portland and Seattle*. Dissertations and Theses. https://doi.org/10.15760/etd.5652.

Mahmoudi, D., Levenda, A. M., & Stehlin, J. G. (2020). Political ecologies of platform urbanism: Digital labor and data infrastructures. In M. Hodson, J. Kasmire, A. McMeekin, J. G. Stehlin, & K. Ward (Eds.), *Urban platforms and the Future City: Transformations in infrastructure, governance, knowledge and everyday life* (pp. 40–52). New York: Routledge.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A. H. (2011). *Big data: The next frontier for innovation, competition, and productivity*.

Mapbox. (2020). *mapbox-gl-js/CHANGELOG.md at main · mapbox/mapbox-gl-js [WWW Document]*. GitHub. URL https://github.com/mapbox/mapbox-gl-js/blob/main/CHANGELOG.md (accessed 7.29.23).

Mapbox. (2023). Create web maps with GL JS | Mapbox [WWW Document]. URL https://www.mapbox.com/mapbox-gljs (accessed 7.29.23).

Mars, M. (2011). *1 million: Class studies.Google web* (chapter 1).

Marx, K. (1992). *Capital: Volume 1: A Critique of political economy (B. Fowkes, trans.; reprint edition)*. Penguin Classics.

McNeill, D. (2015). Global firms and smart technologies: IBM and the reduction of cities. *Transactions of the Institute of British Geographers, 40*, 562–574. https://doi.org/10.1111/tran.12098

Mejias, U. A., & Couldry, N. (2019). *Datafication. Internet Policy Review. 8*.

Mihindukulasuriya, R. (2017). *Esri's annual revenues exceed $1.1 billion [WWW document]*. BW Businessworld. URL http://businessworld.in/article/-Esri-s-Annual-Revenues-Exceed-1-1-Billion-/24-04-2017-116907 (accessed 7.30.23).

Miles, S. (2017). Sex in the digital city: Location-based dating apps and queer urban life. *Gender, Place and Culture, 24*, 1595–1610. https://doi.org/10.1080/0966369X.2017.1340874

Morrison, J. (2020). *Death of an open source business model*. Medium. URL https://joemorrison.medium.com/death-of-an-open-source-business-model-62bc227a7e9b (accessed 7.29.23).

National Labor Relations Board. (2022). *Mapbox vs*. National Labor Relations Board.

Open Source Geospatial Foundation. (2006). *Open source geospatial foundation created to strengthen collaborative development of open geospatial technologies*.

OpenStreetMap. (2021). State of the Map 2021 [WWW Document]. URL http://2021.stateofthemap.org/ (accessed 8.7.23).

O''Reilly Media Incorporated. (2005). *O'Reilly Media, Inc. Launches the 2005 O'Reilly Where 2.0 Conference to Survey the Landscape of Location Technologies*.

O'Sullivan, D. (2006). Geographical information science: Critical GIS. *Progress in Human Geography, 30*(6), 783–791. https://doi.org/10.1177/0309132506071528

Paglen, T. (2009). *Blank spots on the map: The dark geography of the Pentagon's secret world, 1st edition* (edition. ed.). New York: Dutton Adult.

Pope, R. (2016). *Software is politics [WWW document]*. Fast Company. URL https://www.fastcompany.com/3066631/software-is-politics (accessed 6.14.21).

Ramsey, P. (2020). *Mapbox and Morrison [WWW document]*. Paul Ramsey. URL http://blog.cleverelephant.ca/2020/12/mapbox-morrison.html (accessed 7.29.23).

Raymond, E. S. (1998). Homesteading the Noosphere. *First Monday, 3*. https://doi.org/10.5210/fm.v3i10.621

Raymond, E. S. (1999a). The cathedral and the bazaar. *Knowledge, Technology, and Policy, 12*, 23–49.

Raymond, E. S. (1999b). The revenge of the hackers. In *Open sources: Voices from the open source revolution* (pp. 207–219). Beijing; Sebastopol, CA: O'Reilly Media.

Richardson, L. (2018). Feminist geographies of digital work. *Progress in Human Geography, 42*, 244–263. https://doi.org/10.1177/0309132516677177

Ritchie, D. M. (1984). The UNIX system: The evolution of the UNIX time-sharing system. *AT&T Bell Laboratories Technical Journal, 63*, 1577–1593. https://doi.org/10.1002/j.1538-7305.1984.tb00054.x

Ritchie, D. M., Johnson, S. C., Lesk, M. E., & Kernighan, B. W. (1978). The C programming language. *Bell System Technical Journal, 57*, 1991–2019.

Sankaran, S. (2014). *Support for OGC GeoPackage specification in ArcGIS*. ArcGIS Blog. URL https://www.esri.com/arcgis-blog/products/arcgis-desktop/analytics/support-for-ogc-geopackage-specification-in-arcgis/ (accessed 8.7.23).

Shelton, T., Zook, M., & Wiig, A. (2015). The 'actually existing smart city'. *Cambridge Journal of Regions, Economy and Society, 8*, 13–25. https://doi.org/10.1093/cjres/rsu026

Stephens, M. (2013). Gender and the GeoWeb: Divisions in the production of user-generated cartographic information. *GeoJournal, 78*(6), 981–996.

Swartz, J. (2020). *Google becomes third U.S. tech company worth $1 trillion [WWW document]*. MarketWatch. URL https://www.marketwatch.com/story/google-parent-alphabet-joins-1-trillion-in-market-value-for-first-time-2020-01-16 (accessed 6.14.21).

Tanenbaum, A. S. (1987). A UNIX clone with source code for operating systems courses. *ACM SIGOPS Operating Systems Review, 21*, 20–29.

Thatcher, J. (2017). You are where you go, the commodification of daily life through 'location'. *Environment & Planning A*. https://doi.org/10.1177/0308518X17730580, 0308518X17730580.

Thatcher, J., O'Sullivan, D., & Mahmoudi, D. (2016). Data colonialism through accumulation by dispossession: New metaphors for daily data. *Environ Plan D, 34*, 990–1006. https://doi.org/10.1177/0263775816633195

Toftedahl, M., & Engström, H. (2019). A taxonomy of game engines and the tools that drive the industry, in: DiGRA 2019. In *The 12th digital games research association conference, Kyoto, Japan, august, 6–10, 2019*. Digital Games Research Association (DiGRA).

de la Torre, J. (2020). *Our thoughts as MapboxGL JS v2.0 Goes proprietary*. Carto Blog. URL https://carto.com/blog/our-thoughts-as-mapboxgl-js-2-goes-proprietary (accessed 7.29.23).

Turnbull, J., Searle, A., Hartman Davies, O., Dodsworth, J., Chasseray-Peraldi, P., von Essen, E., & Anderson-Elliott, H. (2023). Digital ecologies: Materialities, encounters, governance. *Progress in Environmental Geography, 2*, 3–32.

Tveten, J. (2016). *Code Is Political [WWW Document]*. Vice. URL https://www.vice.com/en/article/3dak5w/code-is-political (accessed 6.14.21).

Wayne, M. (2004). Mode of production: New media technology and the Napster file. *Rethinking Marxism, 16*, 137–154. https://doi.org/10.1080/08935690410001676203

Weber, S. (2004). *The success of open source*. Cambridge Univ Press.

Wendling, A. E. (2009). *Karl Marx on technology and alienation*. Basingstoke [England] ; New York: Palgrave Macmillan.

Williams, S., & Stallman, R. M. (2010). *Free as in freedom (2.0) – Richard Stallman and the free software revolution* (2.0. ed.). Boston, MA: Free Software Foundation.

Yonego, J. T. (2014). *Data is the new oil of the digital economy*. Wired.