Journal of Economic Behavior and Organization 237 (2025) 107109



Contents lists available at ScienceDirect

Journal of Economic Behavior and Organization

journal homepage: www.elsevier.com/locate/jebo



Research paper



Open source software as digital platforms to innovate

Sergio Petralia[1]

Department of Economic Geography, SGPL, Utrecht University, Princetonlaan 8a, 3584 CB Utrecht, The Netherlands

ARTICLE INFO

JEL classification:
O1
R3
T4

Keywords:
Open source software
Patents
Innovation
Digital infrastructure

ABSTRACT

This article provides evidence that organizations routinely leverage Open Source Software (OSS) infrastructure to innovate. It does so by showing that they tend to synchronize both activities, based on an analysis of the timing of patent filings and OSS submissions using a novel database that links two decades of contributions to 98 of the most popular OSS projects with patent filings in the United States for 1556 organizations. These organizations represent the 26.6 % of all patents granted and the 48 % of all OSS contributions to these projects over the period. The results show that this synchronization has intensified over time, predominantly occurs in projects governed by permissive licensing structures, and extends beyond contributions made to open-source projects where the contributing entity holds ownership or control.

1. Introduction

Open Source Software (OSS), which makes its source code publicly accessible for anyone to view, modify, and distribute, has become a cornerstone of products and services in a wide range of industries (Hoffmann et al., 2024), with recent estimates indicating that between 76% and 99% of commercial software codebases include OSS (Synopsys, 2024; Musseau et al., 2022). Android, an OSS mobile operating system, has established itself as the most widely used operating system by web use, occupying 42% of the global market in 2023, a market that, just over a decade ago, was dominated (over 95%) by proprietary versions.[2] GitHub, the most popular OSS code hosting service platform, surpassed 100 million developers in 2023.[3]

A key distinguishing aspect of OSS is the way its development, upgrading, and maintenance are organized. OSS are usually built by a distributed network of individuals and organizations who contribute their skills, knowledge, and resources to a common project or initiative. They use open standards and collaborative tools that promote interoperability, compatibility, and resource-sharing. Participants usually share their original code without the usual exclusionary rights typically associated with the intellectual property rights (IPR) in patent schemes. This particular way of organizing the production process has sparked several puzzling questions since its inception (Lerner and Tirole, 2001, 2002, 2005; Rockett, 2010).

Why would private organizations make original material publicly available for others to use, committing resources into projects they may not have agency over? Prior research has presented open-source contributions as a strategic means of accessing a broad network of specialized individuals, leveraging opportunities for informal knowledge exchange (Henkel, 2006; Dahlander

E-mail address: s.g.petralia@uu.nl.

[1] I am grateful to the anonymous reviewers and the editor for their detailed, valuable, and constructive feedback, which greatly improved the quality of this article. I also thank Enrico Vanino, Joachim Henkel, and Tobias Schmidt for their helpful comments and suggestions. I benefited from comments and feedback received during presentations at the ANET Lab, Utrecht School of Economics, European University Institute, Universidad de la República, Universidad de Valencia, Collegio Carlo Alberto, Università di Firenze, DRUID Conference, University of Stavanger, Bessemer Symposium at the University of Sheffield, Universidad de Buenos Aires, Universidad Torcuato Di Tella, and UNU Merit.

[2] https://gs.statcounter.com/os-market-share#monthly-200901-202303

[3] https://github.blog/2023-01-25-100-million-developers-and-counting/

https://doi.org/10.1016/j.jebo.2025.107109
Received 27 July 2024; Received in revised form 2 June 2025; Accepted 6 June 2025

Available online 24 July 2025
0167-2681/© 2025 The Author. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

and Magnusson, 2008; West and Gallagher, 2006; Johnson, 2002), fostering compatibility, and establishing standards beyond firm boundaries (Lerner and Tirole, 2002, 2005; Bonaccorsi and Rossi, 2003).

A related but relatively understudied perspective is that open-source development holds value for firms that are capable of building innovations on top of the digital infrastructure these projects offer. By granting access to cutting-edge tools, exposing firms to new design paradigms, and enabling collaboration across organizational boundaries, OSS projects create ground for technological experimentation and the development of novel solutions. In turn, this creates an incentive for firms to contribute back to the infrastructure they rely on.

Henkel (2003) shows that firms in the embedded Linux industry strategically leverage the open-source environment to drive innovation by developing and contributing specialized modules or extensions, based on insights from a series of in-depth interviews with companies and industry experts. In addition, Fosfuri et al. (2008) track down OSS product releases by firms in the operating systems and applications niches between 1995 and 2003, showing that complementary assets, such as patents and trademarks, are more likely to release OSS products. Piva et al. (2012) show that entrepreneurial ventures collaborating with the OSS community exhibit superior innovation performance (measured as new or significantly improved software solutions), using a sample of firms collected in 2009 by the general administration of the Emilia–Romagna region of Italy. Relatedly, Cerquera Dussán and Müller (2009) find that OSS adoption correlates with higher product and process innovation value among German firms, while Munir et al. (2018) show that OSS adoption is linked to increased development speed and improved software quality in a Sony Mobile case study.

The empirical evidence on whether firms' innovation strategies benefit from their contributions to, or interactions with, the OSS community is not only limited and fragmented but occasionally contradictory. For instance, Wright et al. (2023) find that participation in OSS activities is positively associated with the establishment of innovative ventures at the national level, whereas Blind et al. (2021) find no clear relationship between different measures of innovation and contributions to OSS using also information collected at the country level.

This article provides evidence that organizations tend to synchronize their internal innovative processes and their contributions to OSS projects, suggesting organizations actively leverage the digital infrastructure provided by OSS to innovate. I take advantage of the timing at which patents are filed and contributions to OSS projects are submitted to show that organizations tend to submit more contributions to the projects mentioned in a patent prior to its filing, as opposed to different benchmark periods. To do so, I create a novel data resource that links patenting activity at the United States Patent and Trademark Office (USPTO) and contributions to 98 of the most popular OSS GitHub repositories for 1556 organizations from 2005 until 2024. These organizations represent a 26.6 % (1,272,367) of all patents granted by the USPTO and a 48 % (4,729,441) of all contributions to these repositories over this period.

The baseline empirical setup relies on comparing the GitHub activity of organizations prior to the two most significant instances in the life of a patent, the day it was filed versus the day it was granted. The main idea behind this strategy is to use the granting date of a patent as a benchmark, relying on the fact that organizations typically do not have control over the schedules or priorities of patent offices, and thus, their GitHub activity prior to that date should not be synchronized or coordinated. The timing of a patent grant is typically subject to a standardized legal procedure that is governed by legal authorities and administrative processes that are exogenous to organizations. Therefore, GitHub contributions to OSS projects mentioned in a patent are undertaken without any knowledge of its granting status prior to its granting date. This independence guarantees the absence of any strategic synchronization of organizations' GitHub activities prior to granting.

On average, organizations tend to submit approximately 18.3% more contributions and do 29.1% more line modifications in the GitHub repositories mentioned in a patent one month prior to its filing date than prior to its granting date. These results are robust to the inclusion of firm-by-year, technology-by-year, OSS project, and seasonal fixed effects, as well as to changes in the time window considered. Similar results are also found using an alternative comparison benchmark, where GitHub activity before the filing of OSS-related patents is compared with the activity preceding the filing of a similar patent that does not reference any OSS project. Notably, these findings also hold when evaluating the activity of individual inventors within organizations. These results provide evidence that organizations tend to synchronize their internal innovative processes and their contributions to the relevant OSS infrastructure, suggesting they actively leverage it to innovate. Additionally, a more detailed analysis reveals that the synchronization of patent and contribution activities has intensified over time, predominantly occurs in projects governed by permissive licensing structures, and extends beyond contributions made to open-source projects where the contributing entity holds ownership or control.

These findings represent a step forward in beginning to understand the underlying dynamics and mechanisms that lead firms to innovate within contemporary technological ecosystems, where traditional boundaries between proprietary and open source development are becoming blurry. They provide evidence that organizations actively leverage the public digital infrastructure offered by OSS to innovate, adopting a strategy that mixes both proprietary and open-source approaches. These results challenge the idea that 'owners of proprietary codes cannot easily allow users to modify their code and customize it without jeopardizing intellectual property rights' (Lerner and Tirole, 2002), while also shedding light on the conditions under which OSS may not threaten, but rather create new opportunities for obtaining IPRs.

These findings contribute to the mostly-theoretical literature studying why and how organizations contribute to OSS (Bessen and Maskin, 2009; Polanski, 2007; Henkel, 2004; Scotchmer, 2010; D'Antoni and Rossi, 2014; Johnson, 2002; Tesoriere and Balletta, 2017). They also offer valuable insights to re-think the existing patent system in light of the skepticism surrounding it (Jaffe and Lerner, 2006; Williams, 2013; Boldrin and Levine, 2013; Moser, 2005, 2013; Gallini, 2002; Nuvolari, 2004), not only in relation to the literature that studies the interaction between OSS licensing and IPR (Lerner and Tirole, 2001; Mustonen, 2003; Suh and Yılmaz, 2019; August et al., 2021) but also more broadly by providing elements that could lead to more effective innovation policy

through, for instance, a deeper understanding of the extent of direct spillovers from OSS activities (Jaffe, 1986; Lerner and Tirole, 2002; West and Gallagher, 2006). Lastly, the findings also relate to the literature that discusses how firms can profit from innovation by leveraging complementary assets or capabilities that enhance the value of an innovation (Teece, 1986) and the importance of firms' abilities to adapt, integrate, and reconfigure internal and external resources to respond to changing environments (Teece et al., 1997) in the context of OSS and the public digital infrastructure they offer.

This article is organized as follows: Section 2 begins by exploring the reasons and mechanisms that drive organizations to innovate using OSS infrastructure, offering practical examples of how these dynamics unfold. Next, Section 3 provides a detailed description of the data sources and the methodology to build a new data resource that links patenting and GitHub activity of organizations. Section 4 presents the main empirical results, while Section 5 concludes.

2. Leveraging OSS as a digital platform to innovate

This section discusses the existing literature that explores the conditions under which OSS projects might enhance an organization's innovation strategy and the incentive structures that would motivate firms to reinvest resources into their development.

2.1. Differentiation through open source innovation

On one hand, Henkel (2004) and Bessen and Maskin (2009) argue that for OSS infrastructure to function as a platform for innovation, it must also enable differentiation. This means OSS should provide organizations with new opportunities to create related yet distinct products, services, or technologies that open up diverse R&D paths. In their models, the stability of cooperative arrangements in which organizations contribute back to OSS projects depends on how quickly imitation could take place. They assume that entry requires investment in specialized capital or that it does not occur instantly. Differentiation gives the original innovator a temporary first-mover advantage, such that competitors will not imitate them easily. By providing a platform for organizations to innovate, OSS projects establish a mechanism that encourages firms to contribute back.

Red Hat, a US-based software company, is a clear example of how this mechanism operates in practice.[4] For over three decades, the company has developed complementary yet differentiated products, services, and technologies by leveraging Linux, the most widely used open-source operating system, while also being one of its most prolific contributors. They do so by offering a range of services including cloud storage, operating system platforms, applications, as well as support, training, and consulting. Its flagship product is Red Hat Enterprise Linux (RHEL), a commercial open source Linux distribution developed for the commercial market. Red Hat has also shown an exceptional ability to leverage the potential of Linux to develop its technologies. It went from producing a handful of patents within its first decade of existence to applying to 135 in 2023 alone (47 of them mentioning Linux in their descriptions).[5] See Henkel (2006) for a similar analysis based on the embedded Linux industry.

2.2. The role of complementary capabilities

On the other hand, Polanski (2007), Scotchmer (2010), Tesoriere and Balletta (2017) and D'Antoni and Rossi (2014) discuss an alternative cooperation mechanism that incentivizes organizations to contribute to OSS infrastructure. When a group of organizations have complementary skills and technologies, such that each of them excels in particular sub-processes and/or have a critical piece of the knowledge necessary for the implementation of a project, cooperating under an open regime can lead to higher profits and greater incentives to invest in both the public and the private components of the innovation process.

In particular, Polanski (2007) shows that in a centipede-type sequential game where innovation is complementary, modular, and exhibits increasing returns to scale, a cooperation system of production may be more robust and stable than a proprietary one. In addition, Henkel (2004) considers a model where contributors are rivals in the market but they create complementary code and show that a regime with compulsory revealing can lead to higher product quality and higher profits than a proprietary one. Similar results have been discussed in Scotchmer (2010), D'Antoni and Rossi (2014), Johnson (2002), Tesoriere and Balletta (2017) and Reisinger et al. (2014).

The 'Hyperledger Fabric' project, a permissioned enterprise-grade blockchain framework, can be considered as an example of such an arrangement.[6] This project is collectively maintained and governed by a diverse group of organizations in sectors like finance, software, and academia, which also possess a disparate set of skills, competencies, and aims.[7] Firms like IBM, Fujitsu, Microsoft, Samsung, or Oracle have been actively leveraging the 'Hyperledger Fabric' project as a platform to innovate while also contributing to its maintenance and development. Particularly interesting is the case of IBM, the top patentee of patents mentioning this project.[8] IBM has been leveraging this digital infrastructure to explore new technological domains, as most patents are in the

[4] In 2012, Red Hat became the first open source technology company to surpass more than 1 billion in revenue. See: https://www.redhat.com/en/about/brand/standards/history.

[5] https://patents.google.com/?assignee=Red.+Hat+Inc.&country=US&before=filing:20231231&after=filing:20230101

[6] Blockchain technologies are part of the broader set of distributed ledger technologies (DLTs), which offer consensus-driven replication, sharing, and synchronization of data. They consist of interlinked records (blocks) that progressively expand as new data is added. These blocks are secured through cryptographic techniques, ensuring the integrity and immutability of the stored information.

[7] See https://en.wikipedia.org/wiki/Hyperledger for more details.

[8] https://patents.google.com/?q=(hyperledger+Fabric)&country=US

category of *cryptographic mechanisms or arrangements for secure communications*,[9] which accounted for only about 1% of its patent portfolio before 2015.

In addition, Fosfuri et al. (2008) discuss an alternative form of complementarity, they argue that firms' IPR portfolios can broaden their capacity to profit from OSS contributions by granting them control over complementary resources, increasing profits on proprietary segments, or displacing competitors (see also Ziedonis, 2004; Mann, 2006; Nagle, 2019).

The next section provides a comprehensive description of the data resources created for this study, along with a step-by-step explanation of the methodology used to link the patenting and OSS activity of organizations. After that, Section 4 outlines the empirical strategy that uses this information to show that there exists a concurrence of organizations' patenting and OSS activities by leveraging the timing at which patents are filed and contributions to OSS are submitted.

3. Data sources

This study draws data from two primary sources. Information about organizations' contributions to OSS projects comes from GitHub, while details about their patenting activity in the United States (US) originates from the United States Patent and Trademark Office (USPTO).

GitHub is the most popular code-hosting service for OSS development.[10] It houses the most extensive collection of code repositories, spanning a variety of different programming languages, frameworks, and libraries across various domains, such as web development, machine learning, data analysis, and mobile applications. One of the key features of GitHub is its smooth integration with *git*, a distributed version control system that facilitates tracking changes in code, managing different versions of a project, and collaborating effectively with others. Through *git*'s branching and merging capabilities, developers can work on different features or bug fixes simultaneously and merge their changes seamlessly. GitHub's public recording of contributions and modifications makes it possible to gain access to the entire development history of any project to study code modification patterns, identify the individuals or teams responsible for specific changes, or understand any aspect of its overall progression.

Given the extensive number of code repositories hosted on GitHub, it is important to note that not all of them hold equal value, with some being forks of other projects or personal code collections. This article focuses on projects that provide a digital infrastructure that could facilitate or enhance innovation, where digital infrastructure refers to a collection of open and shared digital systems that are interoperable, can be reused, and enable the delivery of services or products at a large scale. This perspective extends earlier definitions (G20, 2023; OECD, 2024), which were more narrowly focused on public-sector digital public infrastructure. Table A.10 outlines the types of projects included in the analysis, grouped according to their primary functionality (Compute Resources, Data Management & Storage, Communication, etc.).

The final list of projects was obtained using the following steps: (i) An extensive list of projects (68,029) was gathered from a catalog of curated collections containing information about projects covering a wide range of subjects. This catalog and the collections it contains are typically maintained by the open-source community.[11] (ii) For this list of projects, I collect three metrics that I later use to rank them: the number of forks, watchers, and commits. The number of forks indicates how many users have copied the repository to modify or build upon it, while the number of watchers represents the number of users who actively follow a repository for updates. Commits measure the number of contributions a project has received. Together, these metrics provide insights into a project's popularity, reuse, potential for further development, community interest, perceived value, and overall development activity and maintenance. (iii) I then manually review all projects that rank among the top 100 based on the number of forks, watchers or commits.[12] I evaluate each project according to two criteria: first, whether it meets the earlier definition of digital infrastructure; and second, whether it is identifiable in patent documents. For instance, certain projects like 'Chromium' or 'Flutter' have names that can represent something else in the context of patent documents, making their identification challenging and prone to mistakes. In Appendix A, there is a detailed list of all projects that were included (98 in total), along with a brief description of them and a detailed explanation of how they were chosen. In addition, the data repository of this article contains a database with all the projects that were manually reviewed specifying the reason they were included. The final collection of projects includes a wide variety of domains with a rich longitudinal coverage in terms of their emergence and development over time.

From these projects, I trace every contribution ('commit') made from 2005 until 2024. Each commit represents a significant change made to the codebase, from which I extract information about the author of that contribution, including its name and email address, as well as commit-related information such as how many different files have been changed or the number of lines that have been added and/or deleted. I also extract the commit timestamp, which identifies the time of submission. Appendix A describes the data retrieval process in length.

Table 1 shows the most popular OSS projects along with a concise description of them. The column 'First Commit' indicates the year of the first commit in the sample, while the column 'Commits' provides information on the number of commits for each project from its beginning until 2024. It is worth noting that projects such as TensorFlow (a machine learning framework) have quickly gained relevance despite being relatively recent. The resulting dataset contains 9,780,379 commits in 98 different OSS projects, amounting to approximately 86 million file changes and 6.4 billion line modifications.

[9] Technological class H04L9 in the Cooperative Patent Classification (CPC) system.

[10] GitHub has reported a user base of over 100 million developers as of 2023: https://github.blog/2023-01-25-100-million-developers-and-counting/.

[11] The catalog can be accessed at: https://github.com/sindresorhus/awesome.

[12] After having automatically discarded projects that contain words in their description indicating that they are not digital infrastructure. These stopwords are: game, programming-language, editor, hacktoberfest, code-editor, code editor, education, guide, study plan, list, media center, widget, anime, blog, books, book, collection.

Table 1
Top repositories by number of commits.

	OSSProject	Description	FirstYear	Commits
1	linux	Family of kernels powering a wide range of devices from servers to phones	2005	1,329,904
2	androidos	Software platform that runs smartphones and tablets	2008	1,051,600
3	liferay	Portal platform for developing corporate intranets and digital experiences	2006	848,005
4	mozilla	Web browsers and open web standards	2005	808,603
5	llvm	Collection of tools for compiling and optimizing code in many languages	2005	505,935
6	webkitengine	Browser engine powering many mobile and desktop web browsers	2005	281,077
7	perconacluster	Solution to run MySQL databases across many servers	2008	238,261
8	perconaserver	Enhanced MySQL-compatible database with extra reliability features	2008	226,644
9	perconaxtrabackup	Tool for creating hot backups of MySQL databases	2008	215,040
10	freebsd	UNIX-like operating system known for stability and advanced networking	2005	185,945

Notes: This table presents the top 10 OSS projects in the database, along with a concise description of them generated using ChatGPT 4. The column 'First Commit' indicates the year of the first commit in the sample, while the column 'Commits' provides information on the number of commits for each project until 2024.

Following the collection of commit information, individual commits are attributed to organizations using the web domain of the contributor's email. After that, the patenting and GitHub activity of these organizations is linked based using information that is available on two main sources: (i) Compustat,[13] and (ii) the Whois database.

Compustat is a comprehensive database offering detailed information on publicly traded companies worldwide. It provides unique company identifiers along with information on companies' website domains, their legal names, and their patents granted at the USPTO, among other things. The first step in linking GitHub and the patenting activity of organizations involves matching the web domains of companies listed in Compustat with those extracted from GitHub. After this, I retrieve patents using two approaches. First, I obtain the unique identifier provided by Compustat to retrieve patents granted by the USPTO. The identification of companies' patents in Compustat is only provided for patents granted between 2011 and 2019. To broaden the scope, I employ an additional matching process. This involves retrieving patents outside the 2011–2019 window using the assignee names in USPTO patent documents for those patents already linked to organizations from the previous step. This exact matching process uses non-disambiguated assignee names. Second, I perform an exact matching procedure using company names to identify organizations that have no patents assigned by Compustat during the 2011–2019 period, using the non-disambiguated assignee names provided by USPTO.

The Whois database is a widely used internet directory that contains information on registered owners of domain names or IP addresses, such as the registrant's administrative and billing details, as well as technical contact information. These details are provided by registrars during the domain name registration process. For each web domain identified from GitHub, the registrant's legal name is extracted from the Whois database, when available, and matched against non-disambiguated assignee names within the USPTO records. When a positive match is discovered, indicating that the same organization is the owner of both the domain and a patent, a link is made such that all patents under the same assignee are attributed to the corresponding web domain.

The last step in the retrieval of the patenting information of organizations consists of using the disambiguation of assignees provided by the USPTO to gather the patents that belong to these organizations but that were not identified due to misspellings.

A series of manual checks were conducted on this data to ensure that matches were accurate and that minor inconsistencies or similarities in organization names did not result in false positives or false negatives. For instance, the USPTO disambiguation tends to group organizations within the same identifier based on ownership, which is not ideal since patents from organizations that are not likely to be managing their innovation processes jointly are bundled in practice. To address this, any link where a non-disambiguated name differed by more than three characters from the original Compustat name was removed, and the remaining ones were manually checked. A similar issue arises during the retrieval of organizations' legal names using the Whois protocol, as it is often the case that ultimate owners appear as registrants of their subsidiaries' web domains. These ambiguities were resolved by manually verifying concordances between organizations and web domains. Additionally, the USPTO disambiguation occasionally misses patents from prominent patentees like IBM, Fujitsu, or Broadcom. To mitigate this, I manually reviewed the top hundred patentees excluded from the sample to identify any name discrepancies. A similar approach was applied to the top hundred contributors, verifying whether organizations contribute under alternative web domains (e.g., grouping facebook.com, fb.com, and meta.com under one organization).

After this procedure, a total of 1556 unique organizations with both GitHub and patenting activity were identified. These organizations are responsible for approximately 48.9% of all contributions to the GitHub repositories included in this study. This share is considerably higher at the end of the period. Table 2 presents the top domains in terms of contributions per project.

Data on the patenting activity of organizations is obtained from the USPTO, which offers comprehensive information on all patents granted in the US.[14] They provide details such as the name and location of inventors and/or assignees, as well as the technological domain associated with each invention. The USPTO employs a systematic classification system that categorizes

[13] Available at: https://wrds-www.wharton.upenn.edu/pages/get-data/compustat-capital-iq-standard-poors/compustat/.
[14] This information can be accessed at: https://patentsview.org/download/data-download-tables.

Table 2
Top domains by number of commits in OSS projects.

	Web domain	OSS project	Commits	File Changes
1	google.com	android os	946,803	2,266,292
2	liferay.com	liferay	799,646	4,326,602
3	mozilla.com	mozilla	390,577	6,765,051
4	oracle.com	percona xtrabackup	174,434	1,615,859
5	oracle.com	percona cluster	174,219	1,527,424
6	oracle.com	percona server	170,935	1,508,064
7	apple.com	webkit engine	143,714	2,008,083
8	intel.com	linux	105,043	248,318
9	oracle.com	mysql	92,244	2,547,732
10	google.com	tensorflow	91,358	897,648
11	redhat.com	ceph	90,537	366,540
12	apple.com	llvm	78,252	315,974
13	oracle.com	oracle graal	76,377	979,509
14	google.com	llvm	66,072	325,840
15	redhat.com	linux	62,632	160,511

Notes: This table shows the top 15 web domains (ranked by number of commits) in terms of contributions per project. The columns 'Commits' and 'File Changes' provide information on the number of commits and file changes each domain has submitted for each project until 2024.

Table 3
Top domains by number of patents.

Domains	Patents
samsung.com	128,886
ibm.com	108,838
microsoft.com	38,353
intel.com	36,175
qualcomm.com|quicinc.com	35,257
sony.com|sonymobile.com	34,421
toshiba.co	31,283
apple.com	30,503
google.com|skia.org|tensorflow.org	28,561
hitachi-solutions.com|hitachi.co|hitachi.com	25,861
huawei.com	24,272
ge.com	23,637
fujitsu.com	23,616
micron.com	22,885
hp.com|hpe.com|sgi.com	20,317

Notes: This table presents an overview of the top patentees in the sample using their web domain(s) as identifier.

patents into various technological classes based on the nature of the claimed rights. This classification system provides a structured framework for organizing patents according to their specific fields and areas of innovation.

In this study, I include only patenting information starting from 2005 because it is when GitHub activity predominantly began.[15] Notably, these organizations have collectively generated over 1,255,501 patents from 2005 until 2024, representing approximately 26.6% of all patents granted during this period. Table 3 presents an overview of the top patentees, which includes not only renowned technology giants such as Samsung (samsung.com), Microsoft (microsoft.com), or Intel (intel.com) but also legacy businesses like General Electric (ge.com). This is consistent with the type of organizations found in studies using similar data (Shen, 2023).

The main piece of information connecting both databases is(are) the web domain(s) associated with an organization. Note that while GitHub contributions are made by individuals, meaning that each of them is associated with a single web domain, information about granted patents is arranged at the organization level, which means they can be linked to one or more web domains. During the search and matching process, multiple web domains may have been linked to the same organization because they are registered under the same legal name; for instance, 'sony.com' and 'sonymobile.com' are both registered by 'Sony Corporation'.

Detecting OSS in patent documents

I use the information contained in the text description of patent documents to identify those that mention any of the OSS projects included in the study.[16] The identification of all the projects mentioned in a patent document is later used to differentiate between contributions made to OSS projects that are related (or unrelated) to the patent content around its filing date.

[15] Even though GitHub was founded in 2008, there are commits before that date because older projects like Linux imported and migrated from other version control systems.

[16] I only consider mentions of OSS projects for which the organization has contributed at least once.

Table 4
Patent data descriptive statistics.

	Patent type	
	Mentions an OSS	Others
Observations	68,994	1,203,373
Organizations in Sample	395	1,532
Multi-Inventor Patents	57,962 (84%)	947,454 (78.7%)
Multi-Assignee Patents	649 (0.9%)	56,680 (4.7%)
Foreign-made	15,001 (21.7%)	627,298 (52.1%)
Period: 2005–2024		

Notes: This table provides an overview of the main characteristics of the patent dataset used in this study by type of patent. The column called 'Mentions an OSS' refers to patents that mention the name of any of the projects included in the study. The column called 'Others' refers to patents that do not mention any project.

Table 4 provides an overview of the main characteristics of the patent data that will be used in what follows. Most patents in the sample were produced by teams, which is in line with what we would expect based on prior studies (Jones, 2009; Wuchty et al., 2007). Conversely, patents with multiple assignees occur less frequently. Approximately 5.4% of patents mention at least one of the OSS projects considered in this study. This group is more likely to be created by teams and less likely to involve multiple organizations or originate outside the US.

Final considerations

Several aspects of this data merit careful consideration. It should be clear by now that the focus of this study are organizations that act as both USPTO patentees and OSS contributors, and as such, the analysis that follows is not intended to provide insights into the underlying reasons why firms choose to take on this dual role; rather, it assumes the role is already established. Also, it concentrates solely on USPTO patentees, which means it only incorporates a subset of the global patenting activity, excluding other forms of innovation the not captured in patents (Hall et al., 2014). On the other hand, this is an approach that has been used extensively in economics and innovation studies and can be attributed to the value of USPTO patenting data in addressing global issues. The significance and scale of the US technological market, combined with the consistent and systematic evaluation of patent applications over time, make USPTO patent data a very valuable resource.

Another aspect to take into consideration is the possibility of under-representing organizations' contributions to OSS projects. This is because some employees make contributions anonymously, not using their work email, thereby creating an incomplete portrayal of the actual extent of OSS engagement by these organizations. Additionally, the inclusion of only a selected number of OSS projects may contribute to this under-representation.

Finally, a deliberate choice is made to only consider granted patents, as they represent novel, groundbreaking innovations. This means that even though patenting data is updated until 2024, it may be the case that certain patents filed before that date are not included in the sample, as they might still be pending approval. As we approach the end of the period, this attrition effect intensifies.

4. Empirical analysis

The primary goal of this section is to evaluate to what extent organizations actively leverage OSS infrastructure to innovate by studying the concurrence of patenting and OSS activities. If OSS contributions increase prior to patent filings this may indicate that these activities have required synchronization and simultaneous execution, suggesting that the OSS infrastructure has played an important role in the inventive process, perhaps unlocking specific creative pathways.

This is because innovations that involve interdependent components increase creative challenges, requiring organizations to overcome additional hurdles in specifying, sourcing, and integrating them (Adner and Kapoor, 2010). These challenges can significantly alter their technological and organizational routines, requiring coordination of both their internal innovative processes and their contributions to the OSS infrastructure. In addition, synchronizing OSS and patenting activities can help organizations streamline their innovation workflow by enabling them to develop specific inputs or components internally, rather than waiting for these to become available (Henkel, 2004).

Focusing on the synchronization of OSS and patenting activities presents a well-suited empirical setting, given that such coordination is not assumed nor needed by most existing theories of OSS participation. When the motivation for contributing to OSS is to access a network of specialized individuals, benefit from informal knowledge sharing (Henkel, 2006; Dahlander and Magnusson, 2008; West and Gallagher, 2006; Johnson, 2002), or promote compatibility and establish standards beyond firm boundaries (Lerner and Tirole, 2002, 2005; Bonaccorsi and Rossi, 2003), synchronization of activities is not necessarily required. The same applies to individual inventors and developers. Even if OSS participation leads to external recognition, enhanced performance (Lerner and Tirole, 2005), intrinsic satisfaction (Bitzer et al., 2007), or a sense of community identity (Bagozzi and Dholakia, 2006), these benefits do not necessarily require these activities to be carried out simultaneously or in synchrony with the filing schedules of the firm.

This section evaluates whether organizations increase their contributions to OSS projects that are mentioned in a patent before it is filed. Doing so requires establishing a suitable benchmark for comparison, ideally one that reflects the organization's typical amount of contributions to those projects.

The baseline empirical setup relies on comparing the amount of OSS contributions prior to the two most significant instances in the life of a patent, the day it was filed versus the day it was granted. The granting date of a patent constitutes a good benchmark because organizations do not have control over the schedules or priorities of patent offices. Hence, contributions to OSS projects mentioned in a patent prior to its granting date approximate the typical amount of contributions by the organization. In other words, organizations should not be capable of timing their OSS contributions to projects that are being mentioned in a patent before its granting. The timing of a patent grant is typically subject to a standardized legal procedure, governed by legal authorities and administrative processes, which makes it exogenous to the firm. From the organization's perspective, their contributions to OSS projects related to the patent in question are undertaken without any knowledge of its granting status. Moreover, identifying its exact granting date would be exceptionally difficult due to the substantial variability in the time it takes for a patent to be approved. This uncertainty is largely due to factors such as the backlog of patent applications, the complexity of the technology involved, and the workload of patent examiners.[17] Organizations usually have limited influence or knowledge over these factors.

The main specification, shown in Eq. (1), evaluates whether there are systematic differences in organizations' contributions to the GitHub repositories mentioned in a patent one month before its filing, as opposed to its granting date. More specifically, $C_{f,t,p}$ measures the number of contributions to the OSS projects mentioned in patent p by organization f one month prior to date t, which is either the filing or granting date of the patent. Three different measures of contributions are considered: the logarithm of the number of commits submitted, of the number of code modifications, and of the number of files changed. On the other hand, the main variable of interest, $Filing$, is a dummy variable that takes the value of one if the observation refers to the filing date and zero if it refers to the granting date. Hence, the coefficient of interest (β) measures the extent to which organizations' contributions to OSS projects that are mentioned in a patent tend to be higher prior to its filing than to its granting date.

$$C_{f,t,p} = \beta\, Filing_{t,p} + \theta_{f\,x\,Year} + \delta_{TechxYear} + \omega_{Repos} + \epsilon_{f,t,p} \tag{1}$$

This specification also includes a set of dummy variables that control for unobserved heterogeneity at different levels. On the one hand, firm-by-year fixed effects are included to account for time-varying organizational characteristics that could affect the probability that commit and patenting activity concur in time. Consider, as an example, the potential influence of Research & Development (R&D) yearly budgets. The availability of tighter or wider budgets may have an impact on both the propensity to patent and the willingness to contribute to OSS projects. In the former case, this is because patents are typically the expected output of R&D. In the latter, it may be due to the availability or scarcity of spare resources that can be allocated to OSS contributions. In addition, technology-by-year fixed effects are included to control for the possibility that specific technological developments may be behind an increase in the probability that organizations engage in both activities. For instance, when a technology gains popularity or becomes a trend, this increases the likelihood of organizations mentioning it in a patent and seeking to contribute to it. Finally, repository fixed effects control for the possible heterogeneity in code modification requirements across different repository combinations. It prevents the results from being sensitive to any particular repository combination or any idiosyncratic repository effect. Unless otherwise specified, standard errors (denoted as $\epsilon_{f,t,p}$) are clustered at the level of the fixed effects according to Cameron et al. (2012).

Not all 68,994 patents that mention an OSS project are included in the baseline analysis. This is because patents are excluded whenever there exists another OSS-related patent filing mentioning the same repository by the same organization within a month prior to its granting date. This constraint is crucial to maintain the integrity of the benchmark group, as it avoids an overlapping of filing and granting events which may distort the level of contributions around the benchmark date. After discarding overlapping events, 20,702 patents remain, which represent a 30% of all patents mentioning an OSS in the database. The remaining group is similar to the entire population of patents mentioning an OSS to the extent that both samples contain approximately the same proportion of teams (84%). They differ in that the former group has a disproportionately higher number of multi-assignee organizations (1.8%) and are more likely to be foreign-made (28.6%). Both samples are very similar in terms of their technological distribution, with a 99.6% correlation in technological classes between the two. For each of the 20,702 firm-specific patents, two observations are generated. The first records the number of contributions to the OSS project(s) mentioned in the patent one month before its filing, while the other measures the contributions before its granting. Table B.20 shows the average number of contributions one month before the granting and the filing for these patents. On average, organizations submitted 77.8 commits and 34,053 lines of code in the month leading up to the patent filing, compared to 50.6 commits and 15,082 modifications before the patent was granted.

The results in Table 5 show that there are systematic differences in the OSS activity of firms prior to the filing of a patent as opposed to prior to its granting; organizations tend to submit approximately 18.3% more commits to the GitHub repositories mentioned in the patent prior to its filing date. These contributions tend to be more complex because they contain, on average, a higher number of line modifications. Column (2) shows that organizations tend to make approximately 29% more line modifications prior to filing.

[17] For instance, for the group of patents mentioning an OSS in the sample the average time between filing and granting is approximately 3 years with a standard deviation of approximately 1.5 Years.

Table 5
Concurrence of patent activity and GitHub contributions.

	Commits (in logs) (1)	Modifications (in logs) (2)	File Changes (in logs) (3)
Filing Date	0.183***	0.291***	0.201***
	(0.035)	(0.072)	(0.045)
Period	2005–2024	2005–2024	2005–2024
Dep. Variable Window	One month	One month	One month
Fixed Effects:			
- Firm × Year	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes
Observations	41,404	41,404	41,404
Adjusted R^2	0.742	0.695	0.710

*p < 0.1; **p < 0.05; ***p < 0.01.

Notes: This table contains patent-organization specific regressions where the dependent variables capture different measures of the number of contributions that organizations have submitted within one month prior to the filing or the granting of the patent. *FilingDate* is a dummy variable that takes the value of one when contributions are measured a month prior to the filing and zero when they are measured prior to granting. Control variables include a set of dummies that control for firm-by-year, technology-by-year, and repository combination fixed effects. Standard errors are clustered at the level of the fixed effects according to Cameron et al. (2012).

If we consider Google's contributions to the Android OS, the largest contributor-repository instance included in this study (see Table 2), they explicitly mention 'Android OS' in 28 patents. For these patents, Google submitted an average of 9,486 commits and 2.3 million line modifications in the month before filing, compared to 5,971 commits and 0.7 million modifications in the month before the patent was granted.

These results are robust to several changes in the specification. More specifically, B.21 and B.22 evaluate the sensitivity of the results to minor changes in the way errors are clustered. Tables B.23 & B.24 replicate the results in Table 5, including only patents granted in less than 5 and 3 years, respectively. This ensures that the results do not reflect declining organizational interest as OSS projects mature.[18] In addition, note that seasonal factors may be particularly relevant to explain the concurrence of patenting and commit activity within organizations. For instance, firms and OSS projects might cluster their patenting and commit activities during specific months of the year to meet end-of-year or end-of-quarter deadlines, leading to a convergence of activity that is not necessarily driven by specific organizational needs. Similarly, OSS projects may schedule meetings on specific dates that could coincide with organizations' patent filing routines. Table B.25 shows the robustness of the results to the inclusion of organization-by-month fixed effects to filter out possible seasonal effects. Tables B.30 & B.31 show that results are not sensitive to truncating the sample at either the beginning (starting after 2010) or the end (finishing in 2020). In addition, Table B.26 shows that results are not driven by a disproportionate amount of contributions from top contributors in specific repositories, while Table B.29 shows that the main results are stronger if low-complexity contributions are excluded (i.e. those with descriptions containing the terms 'bug fix' or 'bug fixes'). Finally, Table B.33 shows that the results are robust to the exclusion of organizations with anonymized Whois entries, addressing the concern that such firms might be systematically different because they place greater importance on privacy, which could potentially lead to different contribution behaviors.

Although organizations typically lack full control over the OSS they contribute to, there are instances where they have considerable influence, either by being the most active contributors or by owning the project repository. For example, influential projects such as TensorFlow and PyTorch were originally developed by Google and Facebook, respectively. Table B.27 explores this issue further by evaluating whether there are significant differences between main contributors and the rest. Main contributors are defined as those responsible for more than 50% of total commits, file changes, or line modifications in a given repository (see Appendix A for more details). The results show that the main contributors typically make 65% to 70% more contributions before a patent's filing than before its granting. However, this heterogeneity does not affect the overall findings, as this type of contribution represents a small percentage of the total. This finding can be interpreted as evidence of direct spillovers because it reveals firms primarily leverage OSS they neither own nor actively maintain (Jaffe, 1986; Lerner and Tirole, 2002; West and Gallagher, 2006). Similarly, Table B.28 shows that the results are not sensitive to excluding organizations with potentially different strategic objectives, such as universities and government entities (contributors with web domains ending in '.edu', 'ac.uk', or '.gov').

These results align with the expectation that organizations routinely leverage the OSS infrastructure to innovate. The timely spike in OSS contributions related to the patent content prior to its filing suggests that organizations have synchronized both their internal innovative processes and their contributions to the OSS infrastructure, such that they have coordinated organizational innovation routines to simultaneously execute accompanying changes in OSS components.

The baseline specification has placed an arbitrary emphasis on the period leading up to the filing of patents to evaluate unusual spikes in organizations' contributions to OSS projects. It could be the case, however, that organizations decide to postpone certain modifications that they may consider sensitive after becoming the prior art. This suggests that there may be reasons to expect that

[18] OSS contribution data shows that the opposite is true. Firms included in this study accounted for less than 10% of all commit activity at the beginning of the study period and represent more than 40% in 2024.

Table 6
Concurrence of patent activity and GitHub contributions under an alternative benchmark.

	Commits (in logs) (1)	Modifications (in logs) (2)	File Changes (in logs) (3)
Filing Date	0.19 [0.12, 0.26]	0.3 [0.13, 0.49]	0.2 [0.11, 0.31]
Period	2005–2024	2005–2024	2005–2024
Dep. Variable Window	One month	One month	One month
Fixed Effects:			
- Firm × Year	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes

Notes: This table reports a summary of estimating 100 regressions using different randomly created samples that contrast the GitHub activity of organizations before filing a patent mentioning an OSS with the activity preceding the filing of a similar patent that does not reference any OSS project. It shows the median of all estimated coefficients of interest (β) for each of the OSS contribution measures, along with a summary of the 95% confidence interval of these estimations in brackets. The first value in the bracket represents the lowest value of the estimated lower confidence interval, while the second value corresponds to the highest value of the upper confidence interval. All estimated coefficients are statistically significant, with the lower band of the 95% confidence interval always being greater than zero.

these unusual spikes in OSS contributions do not abruptly stop at the moment of patent filing. Table B.34 presents an alternative set of regressions that explore this hypothesis by evaluating different time windows. The results indicate that contributions tend to spike around the filing date and start to decline as wider time windows are considered.

4.1. An alternative benchmark

An alternative strategy might be to contrast the GitHub activity of organizations before filing an OSS-related patent with the activity preceding the filing of a similar patent that does not reference any of the OSS projects included in this study. In other words, while the baseline setup represents an intra-patent comparison, which evaluates the GitHub activity of organizations along a patent's timeline, an alternative approach could focus on an inter-patent comparison.

Such an approach presents some challenges. On the one hand, any patent used as a benchmark should not only closely resemble the focal patent but should have been applied for after all OSS projects mentioned in it were already established. Otherwise, OSS contributions prior to the filing of the benchmark patent will be trivially zero. In addition, note that for each patent mentioning an OSS, there may be several suitable benchmark patents. This means there are multiple possible ways to match a patent that mentions an OSS project with another that does not.

This section evaluates whether the baseline results are robust to a considerable change in the reference benchmark. The procedure is as follows: first, I create multiple samples with randomly assigned matching configurations between patents that mention an OSS and patents do not, such that: (i) patents that mention an OSS are paired with one randomly selected patent that does not (without replacement); (ii) the control patent should be from the same organization in the same technological class and have the same observable characteristics (i.e. in terms of being a multi-inventor, multi-assignee, and foreign-made); (iii) it has to be filed at most one year prior or after the year the focal patent was filed; (iv) control patents are excluded whenever there exists an OSS-related patent filing by the same firm within a month before its filing; (v) The filing of the control patent should have taken place after the creation of all OSS projects referenced in the focal patent.

For each of these samples, I estimate an analogous model as the one described in Eq. (1), with the difference that now the main variable of interest (OSS) is a dummy variable that takes the value of one if the patent mentions any OSS project and zero otherwise. Hence, the coefficient of interest (β) measures the extent to which organizations' contributions to OSS projects tend to be higher prior to the filing of a patent that mentions them.

Table 6 reports a summary of estimating this model on 100 randomly created samples. It shows the median of all estimated coefficients of interest (β) for each of the OSS contribution measures, along with a summary of the 95% confidence interval of these estimations in brackets. The first value in the bracket represents the lowest value of the estimated lower confidence interval, while the second value corresponds to the highest value of the upper confidence interval. All estimated coefficients are statistically significant, with the lower band of the 95% confidence interval always being greater than zero. These results are similar to those obtained in the baseline estimation; in all instances there are systematic differences between the GitHub activity of organizations prior to the filing of a patent that mentions an OSS project versus those that do not. The amount of commits submitted, lines of code modified, and files changed are higher in the former case than in the latter.

4.2. Placebo test

It is reasonable to expect that organizations would prioritize activities that are relevant to a patent filing prior to its application, potentially postponing non-essential competing tasks. If it is indeed the case that firms increase their OSS contributions prior to the filing of a patent because they need to modify, adapt, or enhance functionalities, this increase in activity should only take place on OSS projects that are mentioned in the patent (or those that are related to them). The purpose of this section is to test whether

Table 7
Concurrence of patent activity and unrelated GitHub contributions.

	Commits (in logs) (1)	Modifications (in logs) (2)	File Changes (in logs) (3)
Filing Date	0.004	−0.056	−0.018
	(0.028)	(0.063)	(0.039)
Period	2005–2024	2005–2024	2005–2024
Dep. Variable Window	One month	One month	One month
Fixed Effects:			
- Firm × Year	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes
Observations	41,404	41,404	41,404
Adjusted R^2	0.835	0.803	0.814

$^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

Notes: This table contains patent-organization specific regressions where the dependent variables capture different measures of the number of contributions to unrelated OSS projects that organizations have submitted within a month prior to the filing of the patent. Projects are flagged as unrelated if they are never mentioned together in a patent. $FilingDate$ is a dummy variable that takes the value of one when contributions are measured one month prior to the filing and zero when they are measured prior to granting. Control variables include a set of dummies that control for firm-by-year, technology-by-year, and repository combination fixed effects. Standard errors are clustered at the level of the fixed effects according to Cameron et al. (2012).

contributions to OSS projects unrelated to the patent's subject matter are lower than usual or exhibit no significant change prior to its filing.

The primary challenge lies in identifying OSS projects that are sufficiently unrelated to one another, as most share common resources such as the same programming languages, contributors, etc. To that purpose, I measure the proximity between projects based on their co-occurrence in patent documents. Projects are flagged as unrelated if they are never mentioned together in a patent (see Appendix A for more details). Therefore, this section evaluates whether the same systematic differences found in Table 5 exist when considering organizations' contributions to unrelated OSS projects, with the expectation that the relationship for these unrelated contributions is weaker, null, or even negative. Table 7 replicates the results of Table 5 using this alternative specification, showing that there is no effect. These findings, or absence of them, shed light on the strategic resource allocation decisions made by firms and individuals, which selectively allocate resources towards OSS contributions that align with their needs or interests.

4.3. On OSS-driven innovation

How does the mechanism that allow firms to leverage OSS infrastructure to innovate look like? Consider the case of a specific OSS innovation, the 'Bfloat16' (brain floating point) format, designed to improve the efficiency of machine learning computations. The detailed analysis of the 'Bfloat16' format provides anecdotal evidence of how this breakthrough, initially introduced through various OSS libraries, catalyzed a wave of contributions from hardware firms seeking to adopt it while also providing them with new possibilities to innovate.

The 'Bfloat16' is a computer number format designed to improve computational efficiency in machine learning by using fewer bits for precision (16 instead of 32) (Kalamkar et al., 2019). This format was developed by Google Brain in 2014, an artificial intelligence research group at Google, and quickly adopted by OSS projects like TensorFlow and PyTorch. Leading semiconductor firms such as Intel and NVIDIA have integrated the Bfloat16 format into their technologies to enhance the efficiency of AI and machine learning workloads, reporting improvements in inference performance of up to 2.85 times.[19]

The complementary nature between the 'Bfloat16' format and the technologies of Intel and NVIDIA has driven both companies to increase their contributions to OSS projects like 'TensorFlow' and 'PyTorch', enhancing the functionality and performance of their respective hardware.[20] In fact, as of 2024 Intel and NVIDIA are among the top contributors to 'TensorFlow' and 'PyTorch', ranking second and third, respectively, in terms of lines of code modified that specifically reference the 'Bfloat16' format.[21] The introduction of the 'Bfloat16' format has also served as a digital platform to innovate, particularly for Intel, that as of October of 2024 has applied for 345 patents that contain the word 'Bfloat16', 125 of which also mention 'TensorFlow' or 'pyTorch'.[22]

Fig. 1 illustrates the concurrent growth in Bfloat16-related patenting and open-source contributions by Intel (panel A) and NVIDIA (panel B). The solid line in both panels represents the cumulative number of contributions to TensorFlow and PyTorch that mention

[19] https://community.intel.com/t5/Blogs/Tech-Innovation/Artificial-Intelligence-AI/Intel-and-Facebook-Accelerate-PyTorch-Performance-with-3rd-Gen/post/1335659

[20] See for instance: https://intel.github.io/neural-compressor/1.14.2/docs/quantization_mixed_precision.html and https://developer.nvidia.com/blog/automatic-mixed-precision-the-turbo-charging-feature-for-faster-ai/.

[21] This is calculated by counting commits that contain the word 'Bfloat16' in their description (message field). This information is available at the data repository of this article.

[22] See the data repository for a full list of these patents.

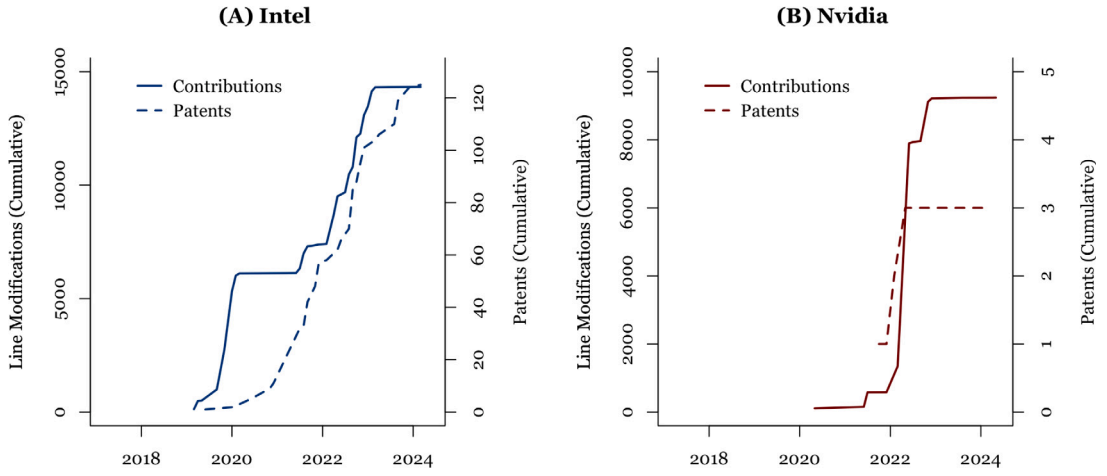


Fig. 1. OSS contributions & patents mentioning the 'Bfloat16' Format.
Notes: This figure displays, on the one hand, the evolution over time of contributions to 'TensorFlow' and 'PyTorch' that mention the 'Bfloat16' format in their descriptions, with data from Intel shown in panel (A) and from NVIDIA in panel (B). This information is represented by solid lines, referenced against the left axis of both graphs. On the other hand, the figure also presents the evolution of patents that mention 'TensorFlow', 'PyTorch', and 'Bfloat16' in their descriptions, shown as dashed lines and referenced against the right axis.

the Bfloat16 format in their descriptions. The dashed lines, in contrast, show the evolution of the number of patent applications referencing TensorFlow, PyTorch, and Bfloat16 in their descriptions. The clear synchrony between the evolution of Bfloat16-related contributions and patenting activities within these firms, coupled with the fact that the introduction of the 'Bfloat16' format was set off through OSS and the technical nature of the associated modifications, reinforces the idea that organizations leverage features of these digital infrastructures to innovate. This idea is also supported by the fact that many inventors involved in these patent applications are also active contributors to the OSS infrastructure. For example, the first author of Intel's patent US12008367B2, which was granted in 2018 and covers systems and methods to improve computational performance using 16-bit floating-point operations, has been an active contributor to TensorFlow since late 2016.[23]

This anecdotal evidence suggests two additional empirical approaches to provide evidence that the core results in this study are innovation-related. On the one hand, if it is indeed the case that organizations use OSS as a platform to innovate, the type of license should significantly influence their strategic behavior. Under permissive licenses, Intel and NVIDIA were able to integrate Bfloat16 into their hardware while tailoring OSS for their specific needs without the obligation to open-source every derivative modification. In contrast, stricter licenses would have mandated that any derivative works also be licensed under the same terms. Such types of license arrangements are likely to discourage firms like Intel or NVIDIA from building proprietary innovations on top of OSS (Lerner and Tirole, 2002; Bonaccorsi and Rossi, 2003; Scotchmer, 2010), as they risk not being able to commercialize those inventions (or parts of them). As a result, firms that want to maintain control over their innovations may be less inclined to contribute under these licenses. Note that this should not be the case if an organization's primary goal is to profit from the spillover effects of open innovation, such as benefiting from a vibrant developer community or attracting talent. In these instances, the obligation to open-source derivative works may be less of a concern. The collaborative benefits and network effects provided by the OSS ecosystem are likely to outweigh the potential downsides of sharing modifications.

Building on this idea, Table 8 reproduces the analysis in Table 5 differentiating patents mentioning repositories with permissive (Apache-2.0, MIT, BSD-3-Clause or no license at all) versus restrictive (GPL-3.0, LGPL-3.0, AGPL-3.0, or GPL-2.0) licenses.[24] Permissive licenses are defined as those that allow firms to modify and incorporate OSS into proprietary projects without requiring them to share their changes. In contrast, restrictive licenses are those that mandate that derivative works remain open-source, limiting firms' ability to commercialize modifications. The results show that the synchronization of contributing and patenting activities occurs only for repositories with permissive licenses. For repositories under restrictive copyleft licenses, there is no statistically significant difference in commits, modifications, or file changes before filing. Overall, these results suggest that the synchronization of commit and patenting activities occurs primarily in repositories that enable firms to innovate.

An alternative approach to providing further evidence that the core results of this study are innovation-related is to focus on the behavior of inventors, who are the primary drivers of innovation within organizations. The number of inventors contributing to OSS nearly doubled throughout the study, rising from 1742 in the first half (2005–2014) to 3253 in the second half (2015–2024).

[23] This 'Bfloat16' related patent is among the earliest by Intel. See https://patents.google.com/patent/US11366663B2/en?oq=US12008367B2 Other patents related to the 'Bfloat16' format, such as US11494163B2, US11869171B2, and the recently granted US12057090B2, also feature inventors who are engaged in OSS development.

[24] Cases mentioning multiple repositories with different license types are excluded from the analysis.

Table 8
Replication of Table 5 for different license types.

	Dependent variable:					
	Commits (in logs)		Modifications (in logs)		File Changes (in logs)	
	Permissive (1)	Restrictive (2)	Permissive (3)	Restrictive (4)	Permissive (5)	Restrictive (6)
Filing Date	0.182***	0.121	0.300***	0.077	0.206***	0.049
	(0.037)	(0.080)	(0.078)	(0.118)	(0.049)	(0.067)
Period	2005–2024	2005–2024	2005–2024	2005–2024	2005–2024	2005–2024
Dep. Variable Window	One month	One month	One month	One month	One month	One month
Fixed Effects:						
- Firm × Year	Yes	Yes	Yes	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes	Yes	Yes	Yes
Observations	39,502	1,732	39,502	1,732	39,502	1,732
Adjusted R^2	0.748	0.717	0.702	0.669	0.717	0.691

*p < 0.1; **p < 0.05; ***p < 0.01.
Notes: This table contains patent-organization specific regressions where the dependent variables capture different measures of the number of contributions that organizations have submitted within a month prior to the filing or the granting of a patent. It differentiates patents mentioning repositories with permissive (Apache-2.0, MIT, BSD-3-Clause, or no license at all) versus restrictive (GPL-3.0, LGPL-3.0, AGPL-3.0, MPL-2.0, LGPL-2.1 or GPL-2.0) licenses. The columns called 'Permissive' contain patents that mention only repositories with permissive licenses while the columns called 'Restrictive' include patents that mentions at least one repository with a restrictive license. *FilingDate* is a dummy variable that takes the value of one when contributions are measured one month prior to the filing and zero when they are measured prior to granting. Control variables include a set of dummies that control for firm-by-year, technology-by-year, and firm-by-repository specific effects. Standard errors are clustered at the level of the fixed effects according to Cameron et al. (2012).

Table 9
Concurrence of patent activity and GitHub contributions of inventors.

	Commits (in logs) (1)	Modifications (in logs) (2)	File Changes (in logs) (3)
Filing Date	0.166***	0.406***	0.228***
	(0.034)	(0.040)	(0.033)
Period	2005–2024	2005–2024	2005–2024
Dep. Variable Window	Six month	Six month	Six month
Fixed Effects:			
- Firm × Year	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes
Observations	3,826	3,826	3,826
Adjusted R^2	0.157	0.151	0.167

*p < 0.1; **p < 0.05; ***p < 0.01.
Notes: This table contains patent-inventor specific regressions where the dependent variables capture different measures of the number of contributions that inventors have submitted within six months prior to the filing or the granting of a patent. *FilingDate* is a dummy variable that takes the value of one when contributions are measured six months prior to the filing and zero when they are measured prior to granting. Control variables include a set of dummies that control for firm-by-year, technology-by-year, and repository combination fixed effects. Standard errors are clustered at the fixed effects level according to Cameron et al. (2012).

These inventor-contributors were identified by precisely matching the first and last names of individuals involved in OSS projects with those listed in patent documents from the same organizations.[25]

Table 9 uses this information to test whether the observed concurrence of patenting and contributing activities also occurs at the individual level, specifically among the actors driving innovation within organizations. It replicates the results of Table 5, focusing on inventors' contributions to the repositories cited in the patents they have filed. There are two key differences compared to the baseline specification. First, it considers a broader, six-month window due to the relative scarcity of contribution instances at the individual level. Second, it includes only patents granted within four years of the filing date. While contributions at the organizational level may have a continuity that is independent of any specific contributor, this is not true at the individual level, where team roles can shift, and individuals may leave the firm over time (Table B.32 shows that results are not sensitive to tightening this time span).

The results show that there are systematic differences in the OSS activity of inventors prior to the filing of a patent compared to before its granting. On average, inventors tend to submit approximately 16.6% more contributions (commits) in the GitHub repositories mentioned in the patent prior to its filing date. These contributions tend to be more complex as they contain, on average, a higher number of line modifications. Column (2) shows that inventors tend to make approximately 40.6 more line modifications prior to filing.

[25] This database is available at the data repository of this article and described in detail on Appendix A.

5. Concluding remarks

This article provides evidence that organizations tend to synchronize their internal innovative processes and their contributions to OSS projects, suggesting they actively leverage the OSS infrastructure to innovate. This synchronization predominantly occurs in projects governed by permissive licensing structures and extends beyond contributions made to open-source projects where the contributing entity holds ownership or control.

This finding represents a step forward in beginning to understand the underlying dynamics and mechanisms that lead firms to innovate within contemporary technological ecosystems, where traditional boundaries between proprietary and open source development are becoming blurry. They provide evidence that organizations routinely adopt innovation strategies that mix both proprietary and open-source schemes, challenging the idea that 'owners of proprietary codes cannot easily allow users to modify their code and customize it without jeopardizing intellectual property rights' (Lerner and Tirole, 2002).

This evidence offers valuable insights into re-evaluating the efficacy of current IPR schemes when it comes to one of their key purposes, namely, the dissemination of information. A deeper understanding of the role of OSS for innovation is essential to address the challenge of identifying an ideal framework for IPR and OSS integration, where we efficiently capitalize on the potential synergies of both systems and reduce the worldwide uncertainties about software patenting (Li, 2019). This is especially relevant in the current context, given the increasing ubiquity and growth of OSS technologies in general and of the Artificial Intelligence (AI) sector in particular, which relies heavily on OSS. A better understanding of this phenomenon could represent an opportunity to address the challenge of the declining productivity of research and the increasing difficulty of generating novel ideas (Bloom et al., 2020), providing policymakers with a valuable tool to create effective strategies to encourage and support innovation, leveraging the role of OSS as a form of public good that benefits society at large (Reisinger et al., 2014; Sacks, 2021; Cowan and Jonard, 2003).

There are relevant aspects that lie beyond the scope of this article and that merit further consideration. One of them is a careful discussion and evaluation of potential tools that could effectively encourage technological change in this context. Previous studies have suggested changes in patenting breadth and duration, the establishment of automatic licensing arrangements, and mechanisms to promote disclosure, among other things. These tools could also be beneficial for OSS projects, contributing to the stability of collaborative efforts among contributors.

This study narrowly focuses on organizations that act as both patentees and OSS contributors, and its main results are based on a sub-sample of all patents that mention OSS projects, which may not correspond to a generalizable target population. However, the data resources provided could be used to offer insights into the underlying reasons why firms choose to take on this dual role and/or to study similar questions on broader target populations. In addition, the analysis concentrates on USPTO patentees, which means it incorporates a subset of the global innovative activity, excluding other forms of innovation that are not captured in patents (Hall et al., 2014) and alternative instruments software firms rely on to innovate, like copyrights, trademarks, or trade secrets (EUIPO, 2020).

Declaration of competing interest

I have nothing to declare.

Appendix A. Data descriptor

The data repository of this article can be accessed at: https://doi.org/10.7910/DVN/UQNVHF. It contains 11 databases and 2 R software codes that replicate the main results of the article. Below, you can find a detailed description of how these databases were constructed and how they are organized.

The database called 'OSS_Contributions.csv' contains information about all contributions to the OSS projects included in this study (listed in Tables A.11 & A.12). This article focuses on projects that provide a digital infrastructure that could facilitate or enhance innovation, where digital infrastructure refers to a collection of open and shared digital systems that are interoperable, can be reused, and enable the delivery of products or services at a large scale. This perspective extends previous definitions (G20, 2023; OECD, 2024), which were more narrowly focused on public sector digital public infrastructure. Table A.10 outlines the types of projects included in the analysis, grouped according to their primary functionality.

The final list of projects for was compiled using the following steps: (i) An extensive list of projects (68,029) was gathered from a catalog of curated collections containing information about projects covering a wide range of subjects, including programming languages, machine learning, cybersecurity, and research tools. This catalog and the collections it contains are typically maintained by the open-source community.[26] (ii) For this list of projects, I collect information about three metrics that I later use to rank them: the number of forks, watchers, and commits. The number of forks indicates how many users have copied the repository to modify or build upon it, while the number of watchers represents the number of users who actively follow a repository for updates. Commits measure the number of contributions a project has received. Together, these metrics provide insights into a project's popularity, reuse, potential for further development, community interest, perceived value, and overall development activity and maintenance. (iii) I then manually review all projects that rank among the top 100 based on the number of forks, watchers or commits, after having automatically discarded projects that contain words in their description indicating that they are not digital infrastructure, such as

[26] The catalog can be accessed at: https://github.com/sindresorhus/awesome.

Table A.10
Types of digital infrastructures by functionality.

Function	Definition
Compute Resources	The software infrastructure that manages and allocates computing power and memory for applications to run. This includes operating systems, virtualization tools, container platforms, and cloud-based compute services that provide environments for deploying and executing software.
Security	The systems and processes that verify identities, establish trust, manage permissions, encrypt, authenticate services, and protect resources.
Data Management & Storage	The tools and services that collect, organize, protect, and preserve data and file systems.
Communication	The protocols and infrastructure that enable different components and services to exchange information, whether synchronously or asynchronously, such as networking layers and service meshes.
Orchestration	The automation frameworks that coordinate how software components are deployed, scaled, configured, and updated, including container schedulers, workflow engines, and pipeline tools.
Integration	The interfaces and gateways that let separate applications link up, share data, and trigger actions automatically such as APIs, event connectors, and adapters.
Monitoring & Observability	The systems and tools that collect, analyze, and visualize telemetry data (such as logs, metrics, and traces) to monitor performance, detect anomalies, diagnose failures, and ensure reliability across digital infrastructure.

'study plan' or 'books'.[27] I assess each project based on two criteria: whether it qualifies as a form of digital infrastructure, such as operating systems, database management frameworks, distributed computing architectures, blockchain technologies, or machine learning libraries, and whether it can be identified within patent documents. For instance, certain projects like 'Chromium' or 'Flutter' have names that can represent something else in the context of patent documents, making their identification challenging and prone to mistakes.

The database called 'OSS_Contributions.csv' contains information about all contributions to the OSS projects included in this study[28] (listed in Tables A.11 & A.12). In addition, the data repository of this article contains a database (described in Table A.13) with all the projects that were manually reviewed, specifying the reason they were included (whether they are infrastructure or not and whether they are searchable). The final collection of projects includes a wide variety of domains with a rich longitudinal coverage in terms of their emergence and development over time.

For each repository, I gather detailed information for all contributions (commits) made from 2005 until 2024, including the commit hash (unique identifier), author email, commit date, and summary measures on the number of file changes, insertions, and deletions implemented. This process involves locally cloning the repositories and using the capabilities of the *git* version control system (a replication script is available in the repository). This approach offers the advantage of being able to retrieve not only the number of commits but also detailed information about their nature, such as the number of modified lines and files. Additionally, it is significantly faster than extracting the same data through the GitHub Application Programming Interface (API) service.

Following the collection of commit information, individual commits are attributed to organizations using the web domain of the contributor's email. To identify and extract web domains from email addresses, the following procedure was implemented:

- The email addresses of commit authors were split using the '@' symbol. For example, the email address *abc@toshiba.co.jp* was split into *abc* and *toshiba.co.jp*.
- The second half of the split email address was further processed to extract the constituent parts of web domains: the root domain and the suffix (e.g., '.com', '.org'). For example, the root domain of *toshiba.co.jp* is *toshiba*, and its suffixes are *.co* and *.jp*.
- The root domain and the first suffix were concatenated to form the domain. In this example, the resulting domain is *toshiba.co*.
- Domains that correspond to common email providers such as 'gmail', 'protonmail', 'hotmail', 'yahoo', or 'outlook' were excluded. The file called *'EmailProviders.txt'* (available at the data repository) contains the list of all email providers that were discarded.[29]

Table A.16 lists and describes all the variables in this database.

The GitHub activity of these organizations (identified based on their website domain) is linked to an organization's name using information that is available on two main sources: (i) Compustat and (ii) the Whois database.

Compustat (available at: https://wrds-www.wharton.upenn.edu) is a comprehensive database offering detailed information on publicly traded companies worldwide. It provides unique company identifiers along with information on companies' website domains, their legal names, and their patents granted at the USPTO, among other things. Web domains extracted from GitHub are then cross-referenced with those listed in Compustat. When an exact match is found, the name of the organization is retrieved.

On the other hand, the Whois database is a widely used internet directory that contains information of registered owners of domain names or IP addresses, such as the registrant's administrative and billing details, as well as technical contact information.

[27] The list of stopwords used is 'education', 'guide', 'study plan', 'list', 'media center', 'widget', 'anime', 'blog', 'books', 'book', 'collection'.
[28] Project names are lowercase and do not contain spaces.
[29] This information can also accessed at: https://gist.github.com/okutbay/5b4974b70673dfdcc21c517632c1f984.

Table A.11
Full list of repositories (1 of 2).

	OSSProject	Description	FirstCommit
1	androidos	A software platform that runs smartphones and tablets	2008
2	ansible	A tool that automates installing and configuring software across servers	2012
3	apacheairflow	A system for scheduling and tracking sequences of data tasks	2014
4	apachecamel	A library for connecting different software systems through reusable patterns	2007
5	apacheflink	A framework for processing large streams of data in real time	2010
6	apachekafka	A messaging system for collecting and distributing event data	2011
7	apachespark	A platform for fast analysis of big datasets in memory	2010
8	apachesuperset	A web app for exploring and visualizing data with charts and dashboards	2015
9	ardupilot	Software that controls drones, planes, and other unmanned vehicles	2010
10	autogpt	An experimental assistant that chains AI tasks to pursue user goals	2023
11	bitcoin	A digital currency that allows peer-to-peer payments without intermediaries	2009
12	caddyserver	A web server that simplifies hosting websites with automatic security	2019
13	caffeframework	A toolkit for training and deploying deep learning models	2013
14	cartodb	A platform to create interactive maps and analyze location data	2011
15	ceph	A system that aggregates many disks into a unified, reliable data store	2005
16	chartjs	A library for drawing simple, responsive charts on web pages	2013
17	cmake	A tool that generates build instructions for compiling software across platforms	2005
18	cockroachdb	A database that spreads data across servers for resilience and scale	2014
19	dubbo	A framework for building services that talk to each other across networks	2011
20	duckdb	A lightweight database optimized for interactive queries on local data	2018
21	elasticsearch	A search engine that lets you find and analyze text quickly	2010
22	esp8266	Firmware and tools for programming low-cost Wi-Fi–enabled microcontrollers	2014
23	ethereum	A blockchain platform for programmable transactions and decentralized apps	2013
24	fastapi	A web framework for building high-performance APIs in Python	2018
25	freebsd	A UNIX-like operating system known for stability and advanced networking	2005
26	godotengine	A toolkit for creating 2D and 3D games with visual scripting support	2013
27	gpt4all	A collection of AI models you can run locally for chat-style tasks	2023
28	gradle	A system for describing and automating how code is compiled and packaged	2008
29	grafana	An interface for monitoring and visualizing metrics from various sources	2013
30	grpc	A protocol for fast, efficient communication between applications in different languages	2014
31	helmcharts	Packaged definitions for installing complex software onto Kubernetes clusters	2015
32	hhvm	A runtime that speeds up PHP and Hack language execution	2010
33	jhipster	A generator that quickly scaffolds modern web applications	2013
34	keras	A user-friendly framework for building and training neural networks	2015
35	kubernetes	A platform that manages containerized applications across clusters of machines	2014
36	laravel	A framework that streamlines building web applications in PHP	2013
37	liferay	A portal platform for developing corporate intranets and digital experiences	2006
38	linux	A family of kernels powering a wide range of devices from servers to phones	2005
39	llvm	A collection of tools for compiling and optimizing code in many languages	2005
40	lobechat	An application interface for chatting with local AI models	2023
41	lodash	A set of helper functions for common tasks in JavaScript	2009
42	MariaDB	A community-driven database system compatible with MySQL	2005
43	mermaidjs	A syntax that turns text definitions into flowcharts and diagrams	2014
44	mongodb	A database storing data as flexible JSON-like documents	2007
45	mozilla	A community behind web browsers and open web standards	2005
46	mrdoob	A username associated with innovative web graphics and multimedia tools	2010
47	mybatis	A framework that maps database queries to objects in code	2010
48	mysql	A widely used system for storing and retrieving structured data	2008
49	neo4j	A database specialized for representing and querying networked relationships	2007

These details are provided by registrars during the domain name registration process. This protocol is then used to retrieve the registrant's legal name for all web domains extracted from GitHub.[30]

Once organization names are linked to web domains, the following procedure is implemented: for web domains with a successful match to Compustat, I retrieve patents using two approaches. First, I obtain the unique identifier provided by Compustat to retrieve patents granted by the USPTO. The identification of companies' patents in Compustat is only provided for patents granted between 2011 and 2019. To broaden the scope, I employ an additional matching process. This involves retrieving patents outside the 2011–2019 window using the assignee names in USPTO patent documents for those patents already linked to organizations from the previous step. This exact matching process uses non-disambiguated assignee names. Second, I perform an exact matching procedure using company names to identify organizations that have no patents assigned by Compustat during the 2011–2019 period, using the non-disambiguated assignee names provided by USPTO.

For each organization name linked to a web domain using the Whois database, I perform an exact matching procedure against non-disambiguated assignee names within the USPTO records. When a positive match is discovered, indicating that the same

[30] When extracting Whois data, several firms employed anonymizer services, which prevented access to their legal names. This records were discarded.

Table A.12
Full list of repositories (2 of 2).

	OSSProject	Description	FirstCommit
50	nestjs	A framework for building scalable server-side applications in TypeScript	2017
51	netdata	A tool that provides real-time performance monitoring and health dashboards	2013
52	nodejs	A runtime for running JavaScript applications outside the browser	2009
53	nuttx	A real-time operating system for resource-constrained embedded devices	2007
54	ollama	A command-line interface for managing and running local AI models	2023
55	openaiwhisper	A speech-to-text system that turns audio into written transcripts	2022
56	opencv	A library for processing and analyzing images and videos	2010
57	openjdk	A toolkit for developing and running Java applications	2007
58	openproject	A collaborative platform for planning and tracking project progress	2006
59	opensearch	A suite for search and analytics, forked from Elasticsearch	2010
60	openwebui	An interface for interacting with local AI models through a browser	2023
61	oraclegraal	A runtime that speeds up Java programs with advanced optimizations	2011
62	osgeogdal	A translator library for reading and writing geospatial data formats	2005
63	paraview	An application for visual exploration of large scientific datasets	2005
64	perconacluster	A solution to run MySQL databases across many servers	2008
65	perconaserver	An enhanced MySQL-compatible database with extra reliability features	2008
66	perconaxtrabackup	A tool for creating hot backups of MySQL databases	2008
67	postgresql	A powerful system for managing and querying structured data	2005
68	powertoys	A collection of small utilities that add convenience features to Windows	2013
69	protobuf	A format for encoding structured data in a compact, language-neutral way	2008
70	pytorch	A framework for building and training neural networks using dynamic graphs	2012
71	qemu	An emulator and virtualizer for running many operating systems on one host	2005
72	rabbitmq	A service that routes messages between applications in various patterns	2007
73	ravendb	A document database with built-in full-text search and distributed clustering	2007
74	reactnative	A toolkit for building mobile apps using JavaScript and native components	2015
75	reactos	An effort to recreate a Windows-compatible operating system from scratch	2005
76	reduxjs	A pattern for managing application state in JavaScript interfaces	2015
77	retroarch	A front end for running video game console emulators	2005
78	rocketmq	A messaging system designed for high throughput and reliable delivery	2013
79	shadcnui	A set of reusable UI components built with modern web standards	2023
80	shadowsocks	A proxy tool that helps bypass network restrictions	2015
81	skia	A graphics library for drawing text, shapes, and images	2006
82	slurm	A scheduler that allocates computing resources on large clusters	2005
83	supabase	A backend service for databases, authentication, and real-time updates	2019
84	sveltejs	A framework that compiles UI components into minimal JavaScript code	2016
85	syncthing	A peer-to-peer tool for keeping folders in sync across devices	2013
86	tailwindcss	A system of utility classes that speeds up styling of web pages	2017
87	tdengine	A database optimized for time-series data from sensors and IoT	2019
88	tensorflow	A platform for building and running machine learning models at scale	2015
89	terraform	A tool for describing and provisioning infrastructure as code	2014
90	trilinos	A collection of algorithms for solving large scientific and engineering problems	2005
91	typicode	A namespace for tools that simplify web development workflows	2013
92	uptimekuma	A monitoring tool that checks website and service availability	2021
93	vitejs	A build tool that provides lightning-fast startup and updates for web apps	2020
94	vuejs	A framework for building interactive user interfaces with declarative templates	2016
95	webkitengine	A browser engine powering many mobile and desktop web browsers	2005
96	wireshark	A network analyzer that captures and inspects data packets	2005
97	zephyros	A real-time operating system for embedded and IoT applications	2015
98	zullip	A team collaboration platform engineered for real-time messaging and coordination	2012

Table A.13
Database containing the list of projects reviewed ('ProjectList.csv').

Variable	Description
Repo	Repository address
ForksCount	Number of Forks
WatchersCount	Number of Watchers
Commits	Number of Commits
LicenseID	Type of License
ForksRanking	Ranking in terms of forks
WatchersRanking	Ranking in terms of watchers
CommitsRanking	Ranking in terms of commits
IsInfrastructure	Flag that identifies whether the repo is considered digital infrastructure
IsSearchable	Flag that identifies whether the repo is searchable in patent documents
Keyword2Search	Keyword to be searched in patent documents
Category	Type of infrastructure based on main functionality
Comment	Description of main functionality

Table A.14
Variables in the GitHub Contributions database ('OSS_Contributions.csv').

Variable	Description
Project	Name of the OSS project (lower cased, without spaces)
CommitHash	Commit hash ID
Date	Contribution timestamp
Year	Year of commit
FilesChanged	Number of files changed
Insertions	Number of line insertions
Deletions	Number of line deletions
WebDomain	Domain extracted from commiter's email

Table A.15
Variables in the patent database ('Domains2Orgs.csv').

Variable	Description
OrganizationName	Raw assignee name, in lowercase and without spaces or special characters.
WebDomains	Web domain(s) associated to the organization

organization is the owner of both the domain and a patent, a link is made such that all patents under the same organization name are attributed to the corresponding web domain.

The last step in the retrieval of the patenting information of organizations consists of using the disambiguation of assignees provided by the USPTO to gather the patents that belong to these organizations but that were not identified due to misspellings.

A series of manual checks has been conducted on this data to ensure that matches are accurate and that minor inconsistencies or similarities in organization names do not result in false positives or false negatives. For instance, the USPTO disambiguation tends to group organizations within the same identifier based on ownership, which is not ideal since patents from organizations that are unlikely to be managing their innovation processes jointly appear bundled in practice. To address this, any link where a non-disambiguated name differed by more than three characters from the original Compustat name was removed, and the remaining ones were manually checked. A similar issue arises during the retrieval of organizations' legal names using the Whois protocol, as it is often the case that ultimate owners appear as registrants of their subsidiaries' web domains. These ambiguities were resolved by manually verifying concordances between organizations and web domains. Additionally, the USPTO disambiguation occasionally misses patents from prominent patentees like IBM, Fujitsu, or Broadcom. To mitigate this, I manually reviewed the top hundred patentees excluded from the sample to identify any name discrepancies. A similar approach was applied to the top hundred contributors, verifying whether organizations contribute under alternative web domains (e.g., grouping facebook.com, fb.com, and meta.com under one organization). The database called 'Domains2Orgs.csv', described in Table A.15, contains the full concordance between web domains and organization names.

The database PatentData.csv, detailed in Table A.16, includes patent information for these organizations dating back to 2005. The datasets described in Tables A.14 and A.16 represent the original data on organizational commit and patenting activity. These were subsequently integrated with information from the USPTO to generate the final databases used in various specifications. To ensure the replicability of the results, the database ReplicationDataMain.csv includes the necessary information to reproduce the main findings and most of the robustness checks presented in this article. A detailed description of this database is provided in Table A.17.

The database named 'Bfloat16Commits.csv' includes all commits that reference the 'Bfloat16' format in their descriptions. It follows the structure outlined in Table A.14, with the addition of a new variable, 'Message', which contains the commit description

Table A.16
Variables in the patent database ('PatentData.csv').

Variable	Description
OrganizationName	Raw assignee name, in lowercase and without spaces or special characters.
patent_id	Patent identifier as provided by USPTO
WebDomains	Web domain(s) associated to the organization

Table A.17
Variables in the database 'ReplicationDataMain.csv'.

Variable	Description
WebDomains	Web domain(s) associated to the organization.
Date	Date of granting or filing.
patent_id	Patent identifier as provided by USPTO.
CPC	Cooperative Patent Classification (CPC) of the patent.
Filing	Dummy variable. One if 'Date' corresponds to the filing date and zero otherwise.
Commits	Commits contributed one month before 'Date' in the repositories mentioned in 'Repo'.
FilesChanged	Files changed one month before 'Date' in the repositories mentioned in 'Repo'.
Modifications	Lines of code modified one month before 'Date' in the repositories mentioned in 'Repo'.
Year	Year of Granting or Filing.
FxY	Firm-by-year ID.
TxY	Technology-by-year ID.
Repos	Repositories mentioned in the patent.
Month	Month of granting or filing.

Table A.18
Variables in the database 'Bfloat16Patents.csv'.

Variable	Description
id	Patent application identified.
title	Patent title.
assignee	Patent assignee(s).
inventor/author	Patent inventor(s).
priority date	Patent priority date.
filing/creation date	Patent filing date.
grant date	Patent grating date.
result link	Link to access the patent in Google patents.

Table A.19
Variables in the database 'ReplicationDataInventors.csv'.

Variable	Description
InventorID	Inventor Identifier.
WebDomains	Web domain(s) associated to the organization.
Date	Date of granting or filing.
patent_id	Patent identifier as provided by USPTO.
Filing	Dummy variable. One if 'Date' corresponds to the filing date and zero otherwise.
Commits	Commits contributed 6 months before 'Date' in the repositories mentioned in 'Repo'.
FilesChanged	Files changed 6 months before 'Date' in the repositories mentioned in 'Repo'.
Modifications	Lines of code modified 6 months before 'Date' in the repositories mentioned in 'Repo'.
Year	Year of Granting or Filing.
TxY	Technology-by-year ID.
FxY	Firm-by-year ID.
Repos	Repositories mentioned in the patent.
Days	Number of days between filing and granting.

provided by the author. The database named 'Bfloat16Patents' contains all patent applications from NVIDIA and Intel that contain the words 'Bloat16' and 'pyTorch' or 'Tensorflow'. This information can be queried and downloaded directly from Google patents.[31] Table A.18 lists and describes all the variables in the database.

Lastly, inventors who are also contributors were identified by exactly matching the first and last names of individuals contributing to OSS projects with those listed in patent documents within organizations. The database called 'ReplicationDataInventors.csv' contains the data to replicate the results in Tables 9 & B.32 and is described in Table A.19.

[31] https://patents.google.com/?q=(Bfloat16)+(pyTorch+OR+TensorFlow)&assignee=Intel+Corp,Nvidia&country=US

Appendix B. Descriptive tables & robustness checks

Table B.20
Average number of contributions one month before filing and granting.

	Before granting	Before filing
Commits	50.6	77.8
Modifications	15,082.9	34,053.7
Files changed	238.6	521.5

Table B.21
Replication of Table 5 with alternative Clustering.

	Commits (in logs) (1)	Modifications (in logs) (2)	File Changes (in logs) (3)
Filing Date	0.183***	0.291***	0.201***
	(0.041)	(0.067)	(0.050)
Period	2005–2024	2005–2024	2005–2024
Dep. Variable Window	One month	One month	One month
Fixed Effects:			
- Firm × Year	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes
Observations	41,404	41,404	41,404
Adjusted R^2	0.742	0.695	0.710

*p < 0.1; **p < 0.05; ***p < 0.01.
Notes: This table contains patent-organization specific regressions where the dependent variables capture different measures of the number of contributions that organizations have submitted within a month prior to the filing or the granting of a patent. $FilingDate$ is a dummy variable that takes the value of one when contributions are measured a month prior to the filing and zero when they are measured prior to granting. Control variables include a set of dummies that control for organization × year, technology-by-year, and repository combination fixed effects. Standard errors are clustered at the level of organization, technological class, year, and repository combination according to Cameron et al. (2012).

Table B.22
Replication of Table 5 with alternative Clustering.

	Commits (in logs) (1)	Modifications (in logs) (2)	File Changes (in logs) (3)
Filing Date	0.183***	0.291***	0.201***
	(0.013)	(0.028)	(0.017)
Period	2005–2024	2005–2024	2005–2024
Dep. Variable Window	One month	One month	One month
Fixed Effects:			
- Firm × Year	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes
Observations	41,404	41,404	41,404
Adjusted R^2	0.742	0.695	0.710

*p < 0.1; **p < 0.05; ***p < 0.01.
Notes: This table contains patent-organization specific regressions where the dependent variables capture different measures of the number of contributions that organizations have submitted within a month prior to the filing or the granting of a patent. $FilingDate$ is a dummy variable that takes the value of one when contributions are measured one month prior to the filing and zero when they are measured prior to granting. Control variables include a set of dummies that control for organization ×'year, technology-by-year, and repository combination fixed effects. Standard errors are clustered at the organization-by-patent level according to Cameron et al. (2012).

Table B.23
Replication of Table 5 only including patents granted in less than 5 years.

	Commits (in logs) (1)	Modifications (in logs) (2)	File Changes (in logs) (3)
Filing Date	0.183***	0.289***	0.197***
	(0.035)	(0.073)	(0.045)
Period	2005–2024	2005–2024	2005–2024
Dep. Variable Window	One month	One month	One month
Fixed Effects:			
- Firm × Year	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes
Observations	38,154	38,154	38,154
Adjusted R^2	0.749	0.700	0.716

*p < 0.1; **p < 0.05; ***p < 0.01.
Notes: This table contains patent-organization specific regressions where the dependent variables capture different measures of the number of contributions that organizations have submitted within a month prior to the filing or the granting of a patent, only for patents granted in less than 5 years. $FilingDate$ is a dummy variable that takes the value of one when contributions are measured two weeks prior to the filing and zero when they are measured prior to granting. Control variables include a set of dummies that control for firm-by-year, technology-by-year, and repository combination fixed effects. Standard errors are clustered at the level of the fixed effects according to Cameron et al. (2012).

Table B.24
Replication of Table 5 only including patents granted in less than 3 years.

	Commits (in logs) (1)	Modifications (in logs) (2)	File Changes (in logs) (3)
Filing Date	0.166***	0.276***	0.180***
	(0.042)	(0.087)	(0.056)
Period	2005–2024	2005–2024	2005–2024
Dep. Variable Window	One month	One month	One month
Fixed Effects:			
- Firm × Year	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes
Observations	25,312	25,312	25,312
Adjusted R^2	0.746	0.698	0.715

*p < 0.1; **p < 0.05; ***p < 0.01.
Notes: This table contains patent-organization specific regressions where the dependent variables capture different measures of the number of contributions that organizations have submitted within a month prior to the filing or the granting of a patent, only for patents granted in less than 3 years. $FilingDate$ is a dummy variable that takes the value of one when contributions are measured one month prior to the filing and zero when they are measured prior to granting. Control variables include a set of dummies that control for firm-by-year, technology-by-year, and repository combination fixed effects. Standard errors are clustered at the level of the fixed effects according to Cameron et al. (2012).

Table B.25
Replication of Table 5 including seasonal fixed effects.

	Commits (in logs) (1)	Modifications (in logs) (2)	File Changes (in logs) (3)
Filing Date	0.175***	0.250***	0.182***
	(0.036)	(0.068)	(0.045)
Period	2005–2024	2005–2024	2005–2024
Dep. Variable Window	One month	One month	One month
Fixed Effects:			
- Firm × Month	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes
Observations	41,404	41,404	41,404
Adjusted R^2	0.679	0.634	0.651

*p < 0.1; **p < 0.05; ***p < 0.01.
Notes: This table contains patent-organization specific regressions where the dependent variables capture different measures of the number of contributions that organizations have submitted within a month prior to the filing or the granting of a patent. $FilingDate$ is a dummy variable that takes the value of one when contributions are measured one month prior to the filing and zero when they are measured prior to granting. Control variables include a set of dummies that control for organization-by-month, technology-by-year, and repository combination fixed effects. Standard errors are clustered at the level of the fixed effects according to Cameron et al. (2012).

Table B.26
Replication of Table 5 the top 3 contributor-repository instances.

	Commits (in logs) (1)	Modifications (in logs) (2)	File Changes (in logs) (3)
Filing Date	0.181***	0.286***	0.198***
	(0.035)	(0.072)	(0.046)
Period	2005–2024	2005–2024	2005–2024
Dep. Variable Window	One month	One month	One month
Fixed Effects:			
- Firm × Year	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes
Observations	41,336	41,336	41,336
Adjusted R^2	0.749	0.698	0.714

*p < 0.1; **p < 0.05; ***p < 0.01.
Notes: This table contains patent-organization specific regressions where the dependent variables capture different measures of the number of contributions that organizations have submitted within a month prior to the filing or the granting of a patent. It excludes contributions to Mozilla by Mozilla, Android OS by Google, and WebKit Engine by Apple to show that results are not driven by a disproportionate amount of contributions from top contributors in specific repositories. $FilingDate$ is a dummy variable that takes the value of one when contributions are measured one month prior to the filing and zero when they are measured prior to granting. Control variables include a set of dummies that control for organization-by-year, technology-by-year, and repository combination fixed effects. Standard errors are clustered at the level of the fixed effects according to Cameron et al. (2012).

Table B.27
Replication of Table 5 comparing main contributors to others.

	Dependent variable:					
	Commits (in logs)		Modifications (in logs)		File Changes (in logs)	
	Rest	Main Contributor	Rest	Main Contributor	Rest	Main Contributor
	(1)	(2)	(3)	(4)	(5)	(6)
Filing Date	0.142***	0.660***	0.233***	0.685***	0.154***	0.772***
	(0.026)	(0.126)	(0.057)	(0.142)	(0.033)	(0.094)
Period	2005–2024	2005–2024	2005–2024	2005–2024	2005–2024	2005–2024
Dep. Variable Window	One month	One month	One month	One month	One month	One month
Fixed Effects:						
- Firm × Year	Yes	Yes	Yes	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes	Yes	Yes	Yes
Observations	40,708	696	40,708	696	40,708	696
Adjusted R^2	0.768	0.913	0.711	0.711	0.732	0.806

*p < 0.1; **p < 0.05; ***p < 0.01.
Notes: This table contains patent-organization specific regressions where the dependent variables capture different measures of the number of contributions that organizations have submitted within a month prior to the filing or the granting of a patent. It shows the result of dividing the sample into 'main contributors' and all other participants, where 'main contributors' are defined as those responsible for more than 50% of total commits, file changes, or line modifications within a given repository. $FilingDate$ is a dummy variable that takes the value of one when contributions are measured one month prior to the filing and zero when they are measured prior to granting. Control variables include a set of dummies that control for firm-by-year, technology-by-year, and firm-by-repository specific effects. Standard errors are clustered at the level of the fixed effects according to Cameron et al. (2012).

Table B.28
Replication of Table 5 including only firms.

	Commits (in logs)	Modifications (in logs)	File Changes (in logs)
	(1)	(2)	(3)
Filing Date	0.180***	0.282***	0.197***
	(0.035)	(0.073)	(0.046)
Period	2005–2024	2005–2024	2005–2024
Dep. Variable Window	One month	One month	One month
Fixed Effects:			
- Firm × Year	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes
Observations	40,396	40,396	40,396
Adjusted R^2	0.744	0.696	0.712

*p < 0.1; **p < 0.05; ***p < 0.01.
Notes: This table contains patent-organization specific regressions where the dependent variables capture different measures of the number of contributions that organizations have submitted within a month prior to the filing or the granting of a patent. It excludes organizations with web domains ending in '.edu','ac.uk', or '.gov'. $FilingDate$ is a dummy variable that takes the value of one when contributions are measured one month prior to the filing and zero when they are measured prior to granting. Control variables include a set of dummies that control for firm-by-year, technology-by-year, month of the year, day of the week, and repository combination fixed effects. Standard errors are clustered at the level of the fixed effects according to Cameron et al. (2012).

Table B.29
Replication of Table 5 excluding contributions mentioning 'bug fix' or 'bug fixes'.

	Commits (in logs) (1)	Modifications (in logs) (2)	File Changes (in logs) (3)
Filing Date	0.259***	0.448***	0.306***
	(0.061)	(0.125)	(0.084)
Period	2005–2024	2005–2024	2005–2024
Dep. Variable Window	One month	One month	One month
Fixed Effects:			
- Firm × Year	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes
Observations	41,404	41,404	41,404
Adjusted R^2	0.770	0.724	0.741

*p < 0.1; **p < 0.05; ***p < 0.01.
Notes: This table contains patent-organization specific regressions where the dependent variables capture different measures of the number of contributions that organizations have submitted within a month prior to the filing or the granting of a patent. It excludes commit activity that contains the words 'bug fix' or 'bug fixes' in their description. $FilingDate$ is a dummy variable that takes the value of one when contributions are measured one month prior to the filing and zero when they are measured prior to granting. Control variables include a set of dummies that control for firm-by-year, technology-by-year, month of the year, day of the week, and repository combination fixed effects. Standard errors are clustered at the level of the fixed effects according to Cameron et al. (2012).

Table B.30
Replication of Table 5 until 2020.

	Commits (in logs) (1)	Modifications (in logs) (2)	File Changes (in logs) (3)
Filing Date	0.141***	0.187***	0.147***
	(0.034)	(0.058)	(0.039)
Period	2005–2020	2005–2020	2005–2020
Dep. Variable Window	One month	One month	One month
Fixed Effects:			
- Firm × Year	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes
Observations	21,628	21,628	21,628
Adjusted R^2	0.625	0.598	0.598

*p < 0.1; **p < 0.05; ***p < 0.01.
Notes: This table contains patent-organization specific regressions where the dependent variables capture different measures of the number of contributions that organizations have submitted within a month prior to the filing or the granting of a patent. It excludes patenting activity after the year 2020 to test for possible truncation effects generated by the delay in the granting process. $FilingDate$ is a dummy variable that takes the value of one when contributions are measured one month prior to the filing and zero when they are measured prior to granting. Control variables include a set of dummies that control for firm-by-year, technology-by-year, month of the year, day of the week, and repository combination fixed effects. Standard errors are clustered at the level of the fixed effects according to Cameron et al. (2012).

Table B.31
Replication of Table 5 after 2010.

	Commits (in logs) (1)	Modifications (in logs) (2)	File Changes (in logs) (3)
Filing Date	0.188***	0.304***	0.208***
	(0.035)	(0.072)	(0.046)
Period	2010–2024	2010–2024	2010–2024
Dep. Variable Window	One month	One month	One month
Fixed Effects:			
- Firm × Year	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes
Observations	39,527	39,527	39,527
Adjusted R^2	0.746	0.698	0.713

*p < 0.1; **p < 0.05; ***p < 0.01.
Notes: This table contains patent-organization specific regressions where the dependent variables capture different measures of the number of contributions that organizations have submitted within a month prior to the filing or the granting of a patent. It excludes patenting activity before the year 2010 to test for possible truncation effects generated by early activity in GitHub, which was created in 2008. $FilingDate$ is a dummy variable that takes the value of one when contributions are measured one month prior to the filing and zero when they are measured prior to granting. Control variables include a set of dummies that control for firm-by-year, technology-by-year, month of the year, day of the week, and repository combination fixed effects. Standard errors are clustered at the level of the fixed effects according to Cameron et al. (2012).

Table B.32
Replication of Table 9 including patents granted within 2 years.

	Commits (in logs) (1)	Modifications (in logs) (2)	File Changes (in logs) (3)
Filing Date	0.130***	0.340***	0.293***
	(0.041)	(0.043)	(0.065)
Period	2005–2024	2005–2024	2005–2024
Dep. Variable Window	Six month	Six month	Six month
Fixed Effects:			
- Firm × Year	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes
Observations	1,216	1,216	580
Adjusted R^2	0.143	0.135	0.185

*p < 0.1; **p < 0.05; ***p < 0.01.
Notes: This table contains patent-inventor specific regressions where the dependent variables capture different measures of the number of contributions that inventors have submitted within six months prior to the filing or the granting of a patent. $FilingDate$ is a dummy variable that takes the value of one when contributions are measured six months prior to the filing and zero when they are measured prior to granting. Control variables include a set of dummies that control for firm-by-year, technology-by-year, and repository combination fixed effects. Standard errors are clustered at the fixed effects level according to Cameron et al. (2012).

Table B.33
Replication of Table 5 for organizations not found in Whois.

	Commits (in logs) (1)	Modifications (in logs) (2)	File Changes (in logs) (3)
Filing Date	0.154***	0.272***	0.171***
	(0.032)	(0.069)	(0.042)
Period	2005–2024	2005–2024	2005–2024
Dep. Variable Window	One month	One month	One month
Fixed Effects:			
- Firm × Year	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes
Observations	35,290	35,290	35,290
Adjusted R^2	0.772	0.711	0.729

*p < 0.1; **p < 0.05; ***p < 0.01.
Notes: This table contains patent-organization specific regressions where the dependent variables capture different measures of the number of contributions that organizations have submitted within a month prior to the filing or the granting of a patent. It includes only organizations that were unidentifiable through the Whois protocol, suggesting they have anonymized their domain registration details. $FilingDate$ is a dummy variable that takes the value of one when contributions are measured one month prior to the filing and zero when they are measured prior to granting. Control variables include a set of dummies that control for firm-by-year, technology-by-year, month of the year, day of the week, and repository combination fixed effects. Standard errors are clustered at the level of the fixed effects according to Cameron et al. (2012).

Table B.34
Replication of Table 5 using different time windows.

	One month before (1)	Two weeks before (2)	One week before (3)	One week after (4)	Two weeks after (5)	One month after (6)
Commits (in logs)	0.183***	0.195***	0.185***	0.182***	0.179***	0.162***
	(0.035)	(0.029)	(0.026)	(0.026)	(0.029)	(0.033)
Modifications (in logs)	0.291***	0.330***	0.334***	0.341***	0.297***	0.244***
	(0.072)	(0.058)	(0.050)	(0.047)	(0.056)	(0.070)
File Changes (in logs)	0.201***	0.221***	0.219***	0.220***	0.224***	0.176***
	(0.045)	(0.037)	(0.034)	(0.033)	(0.041)	(0.043)
Period	2005–2024	2005–2024	2005–2024	2005–2024	2005–2024	2005–2024
Fixed Effects:						
- Firm × Year	Yes	Yes	Yes	Yes	Yes	Yes
- Technology × Year	Yes	Yes	Yes	Yes	Yes	Yes
- Repository Combination	Yes	Yes	Yes	Yes	Yes	Yes
Observations	41,404	41,404	41,404	41,404	41,404	41,404

*p < 0.1; **p < 0.05; ***p < 0.01.
Notes: This table replicates the results of Table 5 using different time windows. Rows represent different measures of the number of contributions that organizations have submitted, while columns specify the time window considered. Cells in this table contain the estimated coefficient replicating the specification used for Table 5. Control variables include a set of dummies that control for firm-by-year, technology-by-year, and repository combination fixed effects. Standard errors are clustered at the level of the fixed effects according to Cameron et al. (2012). For columns 4–6, contributions are considered before the granting date instead of after it. This is because firms could alter (and thus synchronize) their contributing activity after learning that a patent has been granted, for example, by submitting code that may be sensitive.

Data availability

I have created a public repository with all the data and the codes to generate it. I have also provided a replication package.

References

Adner, R., Kapoor, R., 2010. Value creation in innovation ecosystems: How the structure of technological interdependence affects firm performance in new technology generations. Strat. Manag. J. 31 (3), 306–333.

August, T., Chen, W., Zhu, K., 2021. Competition among proprietary and open-source software firms: The role of licensing in strategic contribution. Manag. Sci. 67 (5), 3041–3066.

Bagozzi, R.P., Dholakia, U.M., 2006. Open source software user communities: A study of participation in linux user groups. Manag. Sci. 52 (7), 1099–1115.

Bessen, J., Maskin, E., 2009. Sequential innovation, patents, and imitation. Rand J. Econ. 40 (4), 611–635.

Bitzer, J., Schrettl, W., Schröder, P.J., 2007. Intrinsic motivation in open source software development. J. Comp. Econ. 35 (1), 160–169.

Blind, K., Böhm, M., Grzegorzewska, P., Katz, A., Muto, S., Pätsch, S., Schubert, T., 2021. The Impact of Open Source Software and Hardware on Technological Independence, Competitiveness and Innovation in the EU Economy 10. Final Study Report, European Commission, Brussels, 430161.

Bloom, N., Jones, C.I., Van Reenen, J., Webb, M., 2020. Are ideas getting harder to find? Am. Econ. Rev. 110 (4), 1104–1144.

Boldrin, M., Levine, D.K., 2013. The case against patents. J. Econ. Perspect. 27 (1), 3–22.

Bonaccorsi, A., Rossi, C., 2003. Why open source software can succeed. Res. Policy 32 (7), 1243–1258.

Cameron, A.C., Gelbach, J.B., Miller, D.L., 2012. Robust inference with multiway clustering. J. Bus. Econom. Statist..

Cerquera Dussán, D., Müller, B., 2009. Open Source, ICT Infrastructure and Firm Performance. Tech. rep., ZEW Discussion Papers.

Cowan, R., Jonard, N., 2003. The dynamics of collective invention. J. Econ. Behav. Organ. 52 (4), 513–532.

Dahlander, L., Magnusson, M., 2008. How do firms make use of open source communities? Long Range Plan. 41 (6), 629–649.

D'Antoni, M., Rossi, M.A., 2014. Appropriability and incentives with complementary innovations. J. Econ. Manag. Strat. 23 (1), 103–124.

EUIPO, 2020. Open source software in the European Union. Tech. Rep. TB-03-20-304-EN-N, European Union Intellectual Property Office, http://dx.doi.org/10.2814/866548, Reproduction is authorised provided the source is acknowledged.

Fosfuri, A., Giarratana, M.S., Luzzi, A., 2008. The penguin has entered the building: The commercialization of open source software products. Organ. Sci. 19 (2), 292–305.

G20, 2023. G20 New Delhi leaders' declaration. URL https://www.consilium.europa.eu/media/66739/g20-new-delhi-leaders-declaration.pdf.

Gallini, N.T., 2002. The economics of patents: Lessons from recent US patent reform. J. Econ. Perspect. 16 (2), 131–154.

Hall, B., Helmers, C., Rogers, M., Sena, V., 2014. The choice between formal and informal intellectual property: A review. J. Econ. Lit. 52 (2), 375–423.

Henkel, J., 2003. Software development in embedded Linux—Informal collaboration of competing firms. In: Wirtschaftsinformatik 2003/Band II: Medien—Märkte—Mobilität. Springer, pp. 81–99.

Henkel, J., 2004. The jukebox mode of innovation-a model of commercial open source development. Available at SSRN 578142.

Henkel, J., 2006. Selective revealing in open innovation processes: The case of embedded Linux. Res. Policy 35 (7), 953–969.

Hoffmann, M., Nagle, F., Zhou, Y., 2024. The value of open source software. In: Harvard Business School Strategy Unit Working Paper (24-038).

Jaffe, A.B., 1986. Technological opportunity and spillovers of R & D: Evidence from firms' patents, profits, and market value. Am. Econ. Rev. 984–1001.

Jaffe, A.B., Lerner, J., 2006. Innovation and its discontents. Capital. Soc. 1 (3).

Johnson, J.P., 2002. Open source software: Private provision of a public good. J. Econ. Manag. Strat. 11 (4), 637–662.

Jones, B.F., 2009. The burden of knowledge and the "death of the renaissance man": Is innovation getting harder? Rev. Econ. Stud. 76 (1), 283–317.

Kalamkar, D., Mudigere, D., Mellempudi, N., Das, D., Banerjee, K., Avancha, S., Vooturi, D.T., Jammalamadaka, N., Huang, J., Yuen, H., et al., 2019. A study of BFLOAT16 for deep learning training. arXiv preprint arXiv:1905.12322.

Lerner, J., Tirole, J., 2001. The open source movement: Key research questions. Eur. Econ. Rev. 45 (4–6), 819–826.

Lerner, J., Tirole, J., 2002. Some simple economics of open source. J. Ind. Econ. 50 (2), 197–234.

Lerner, J., Tirole, J., 2005. The economics of technology sharing: Open source and beyond. J. Econ. Perspect. 19 (2), 99–120.

Li, Y., 2019. The current dilemma and future of software patenting. IIC- Int. Rev. Intellect. Prop. Compét. Law 50 (7), 823–859.

Mann, R.J., 2006. Commercializing open source software: Do property rights still matter. Harv. JL & Tech. 20, 1.

Moser, P., 2005. How do patent laws influence innovation? Evidence from nineteenth-century world's fairs. Am. Econ. Rev. 95 (4), 1214–1236.

Moser, P., 2013. Patents and innovation: Evidence from economic history. J. Econ. Perspect. 27 (1), 23–44.

Munir, H., Linåker, J., Wnuk, K., Runeson, P., Regnell, B., 2018. Open innovation using open source tools: A case study at Sony Mobile. Empir. Softw. Eng. 23, 186–223.

Musseau, J., Meyers, J.S., Sieniawski, G.P., Thompson, C.A., German, D., 2022. Is open source eating the world's software? Measuring the proportion of open source in proprietary software using Java binaries. In: Proceedings of the 19th International Conference on Mining Software Repositories. pp. 561–565.

Mustonen, M., 2003. Copyleft—the economics of Linux and other open source software. Inf. Econ. Policy 15 (1), 99–121.

Nagle, F., 2019. Open source software and firm productivity. Manag. Sci. 65 (3), 1191–1215.

Nuvolari, A., 2004. Collective invention during the British Industrial Revolution: The case of the Cornish pumping engine. Camb. J. Econ. 28 (3), 347–363.

OECD, 2024. Digital Public Infrastructure for Digital Governments. OECD Public Governance Policy Papers 68, OECD Publishing, Paris, http://dx.doi.org/10.1787/ff525dc8-en.

Piva, E., Rentocchini, F., Rossi-Lamastra, C., 2012. Is open source software about innovation? Collaborations with the open source community and innovation performance of software entrepreneurial ventures. J. Small Bus. Manag. 50 (2), 340–364.

Polanski, A., 2007. Is the General Public Licence a rational choice? J. Ind. Econ. 55 (4), 691–714.

Reisinger, M., Ressner, L., Schmidtke, R., Thomes, T.P., 2014. Crowding-in of complementary contributions to public goods: Firm investment into open source software. J. Econ. Behav. Organ. 106, 78–94.

Rockett, K., 2010. Property rights and invention. In: Handbook of the Economics of Innovation, Vol. 1. Elsevier, pp. 315–380.

Sacks, M., 2021. Incentives for the over-provision of public goods. J. Econ. Behav. Organ. 191, 197–213.

Scotchmer, S., 2010. Openness, open source, and the veil of ignorance. Am. Econ. Rev. 100 (2), 165–171.

Shen, L., 2023. Does working from home work? A natural experiment from lockdowns. Eur. Econ. Rev. 151, 104323.

Suh, J., Yılmaz, M., 2019. Economics of open source technology: A dynamic approach. Dyn. Games Appl. 9, 254–280.

Synopsys, 2024. Open Source Security and Risk Analysis Report. Tech. rep., Synopsys.

Teece, D.J., 1986. Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. Res. Policy 15 (6), 285–305.

Teece, D.J., Pisano, G., Shuen, A., 1997. Dynamic capabilities and strategic management. Strat. Manag. J. 18 (7), 509–533.

Tesoriere, A., Balletta, L., 2017. A dynamic model of open source vs proprietary R&D. Eur. Econ. Rev. 94, 221–239.

West, J., Gallagher, S., 2006. Challenges of open innovation: The paradox of firm investment in open-source software. R & D Manage. 36 (3), 319–331.

Williams, H.L., 2013. Intellectual property rights and innovation: Evidence from the human genome. J. Political Econ. 121 (1), 1–27.

Wright, N.L., Nagle, F., Greenstein, S., 2023. Open source software and global entrepreneurship. Res. Policy 52 (9), 104846.

Wuchty, S., Jones, B.F., Uzzi, B., 2007. The increasing dominance of teams in production of knowledge. Science 316 (5827), 1036–1039.

Ziedonis, R.H., 2004. Don't fence me in: Fragmented markets for technology and the patent acquisition strategies of firms. Manag. Sci. 50 (6), 804–820.