frontiers in Human Dynamics

ORIGINAL RESEARCH
published: 25 October 2021
doi: 10.3389/fhumd.2021.642556

What Open Source Software Research Can Teach Us About Public Blockchain(s)?—Lessons for Practitioners and Future Research

*Juho Lindman**

University of Gothenburg, Gothenburg, Sweden

Peer-to-peer governance of blockchain technology reemerges a number of interesting practical and theoretical questions. This article aims to bridge current research on blockchain technology to earlier research on open source software (OSS) and to suggest a number of concepts from OSS research that are useful in discussing governance of blockchain systems. Thus, the purpose of this article is to provide a theoretically oriented review of some of the earlier concepts and discuss their applicability in a novel context. Bridging these extending literatures and concepts accelerates theoretical development in the area of governance of technology, opening fertile avenues for future research and offering a variety of insights to both practitioners.

Keywords: blockchain, public blockchain, open source, OSS 2.0, forking

OPEN ACCESS

Edited by:
Primavera De Filippi,
UMR7106 Centre d'Etudes et de Recherches de Sciences Administratives et Politiques (CERSA), France

Reviewed by:
Thomas Hardjono,
Massachusetts Institute of Technology, United States
Jason Potts,
RMIT University, Australia
Sinclair Davidson,
RMIT University, Australia

Correspondence:
Juho Lindman
juho.lindman@ait.gu.se

Specialty section:
This article was submitted to Social Networks, a section of the journal Frontiers in Human Dynamics

Received: *16 December 2020*
Accepted: *25 February 2021*
Published: *25 October 2021*

Citation:
Lindman J (2021) What Open Source Software Research Can Teach Us About Public Blockchain(s)?—Lessons for Practitioners and Future Research. Front. Hum. Dyn. 3:642556. doi: 10.3389/fhumd.2021.642556

INTRODUCTION

Research centered around specific emerging technologies too easily focuses on hyping the novelty of said innovations at the expense of connecting to earlier relevant research. This omission may be augmented if the novel innovation is technically complex and if the earlier relevant research is scattered across several neighboring scientific disciplines.

This is the case in blockchain technology research, one of the most hyped novel innovations in recent years. Blockchain[1] is a technology that relies on a tamper-resistant peer-to-peer network to provide decentralized storage (a shared state) and computing across a computer network. The most well-known blockchain technology application is the bitcoin payment system, protocol, and cryptocurrency (Beck et al., 2017; Lindman et al., 2017, 2020; Xu et al., 2017).

Especially, earlier research on OSS[2] holds untapped potential for the theoretical developments of blockchain: forks, community governance, developer community, and so forth. Many of the terms employed to discuss blockchain technology have their conceptual roots in OSS development. However, research that cataloged these developments from the mid-nineties onward has been mainly omitted from blockchain-related discussions for a number of reasons. Apparently, there is a disconnect between blockchain technology discussions from earlier research on open software development practice. Proponents of blockchain often seem almost unaware of related theoretical discussions presented in earlier research or choose for other purposes not to draw parallels to those discussions.

[1] For a more thorough discussion around blockchain technology, see section From OSS 2.0 to Blockchains (2006–2020).
[2] WOSS is referred initially to OSS as any software licensed under an OSI-accepted license, but see a more thorough discussion on the definition of "OSS" in section Background.

Thus, the main question addressed in this paper is what can OSS research teach us about public blockchain(s). Answers to this question are divided conceptually into three parts:

- 1) OSS 2.0 research,
- 2) research on governance of community development, and
- 3) research on forking.

These insights are interesting for researchers and practitioners working in this area. Note that OSS research is not the only discipline that could offer interesting insights of earlier research—there is plenty of similar work that can be carried out related to blockchain.

This paper is entirely theoretical; it draws on earlier research work in this area. Blockchains are a multifaceted phenomenon, thus the discussions in this article are limited to the two most well-known public blockchain ecosystems: Bitcoin and Ethereum.

The paper is structured as follows. First, this section provides a short introduction to the themes. Section Background introduces relevant definitions and theoretical directions. Then, Section Discussion discusses the introduced concepts in three different areas: OSS 2.0, community governance, and forking. Section Conclusion concludes the article after providing practical implications and limitations.

BACKGROUND

The clash between the commercialization of software and the hacker subculture can be traced back to the entry of the microcomputer and IBM's decision to unbundle hardware and software (for history, see, for example, Weber, 2004). Openness in software production also has a long and rich history that goes back to the nature of software as a commercial product/service, including discussions around concepts such as packaged software (Xu and Brinkkemper, 2007), free software (Stallman, 2002), and commons-based peer production of digital goods (Benkler, 2006).

From Free Software to OSS (1980s−1999)

Discussions regarding open and openness gained a significant boost in 1999, when a group of OSS proponents, including Raymond (1999) and Perens (1999), decided to increase OSS (formerly known as free software) credibility in the commercial context. Raymond published a book titled *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. The book's goal (and other related efforts) was to enroll the twin audience of OSS enthusiasts and business managers behind the same concept and to legitimize them (Fitzgerald, 2006; Marsan et al., 2020). This goal was attained, although these discursive maneuvers also resulted in plenty of critiques, especially from preceding free software advocates (see, for example[3], Stallman, 2002; Szczepanska et al., 2005; more recently, the discussion continues in Schlagwein et al., 2017). This stream of literature on OSS led to a fertile, though contested, research terrain tightly coupled with commercial companies' continuously increasing industrial OSS engagement (Weber, 2004).

As the development continued, new tools and practices related to OSS changed [for example, from Source Forge to GitHub(s) and intranet-based OSS implementations]. Ultimately, even the most critical voices found themselves engaging with OSS. For example, one of the most vocal early critics, Microsoft, joined the Linux Foundation on November 16, 2016[4] In sum, this redefinition strategy can be seen as successful: OSS became mainstream.

From OSS to OSS 2.0 (1999–2006)

OSS practices describe the logic used to organize software production in an OSS community. These would include the shared technical infrastructure used in the development, reward structures for contributions, supporting work, task composition and assignment, and knowledge transfer. Early practices usually relied on primary communication via email, open availability of code from the source repository, some kind of web presence of the project and community (initially Sourceforge) use of versioning control and issue trackers.

OSS is "not a precise term" (Gacek and Arief, 2004)—it was often used as a buzzword that had several meanings, leaving some room for maneuvering and local renegotiation when it was used (Swanson and Ramiller, 1997). Despite providing some common reference points for practitioners, they encountered this fluidity in organizational efforts to change local software production practices in commercial organization. Thus, the resulting community discourse around the phenomenon's terms and philosophical underpinnings meant that the term "OSS" remained an umbrella term to include a number of approaches: in other words, unnecessarily fluid and elusive for many business managers (Hauge et al., 2010). One of the key actors, Open Source Initiative (OSI[5]), stepped in to provide a standard that would help reduce legal uncertainty and provide a definition to which this paper also adheres.

OSS's success was observed in commercial organization and many initiatives started to incorporate some of the ideas of open development communities to their own organizations. These concepts for internal software production included Corporate Source (Dinkelacker et al., 2002) and Inner Source (Van Der Linden et al., 2009).

One key moment for OSS was when Fitzgerald (2006) published a paper to discuss how OSS had evolved when it mainstreamed between 1998 and 2006. The concept of OSS 2.0 was the main theoretical concept pushed forward in the article. The article defined it in this way: "OSS has

[3] For a historical formulation of the argument, see https://www.gnu.org/philosophy/free-software-for-freedom.en.html

[4] It is worth noting that, while Microsoft is often perceived to be suspect of early OSS efforts, the company has a long and complicated history with OSS. For example, MIT Athena/MIT Kerberos software source code was available already in 1986. Microsoft was a founding member of MIT Kerberos Consortium established in December 2007. Even before this, Microsoft first incorporated the MIT Kerberos OSS into Windows NT in 1999 (now called Windows Server). Here is an initial description of this effort (from 1988): https://users.soe.ucsc.edu/$\sim$sbrandt/221/Papers/Security/steiner-usenix88w.pdf

[5] https://opensource.org/licenses

undergone a significant transformation from its free software origins to a more mainstream, commercially viable form" that would be "a harbinger of the end to the current dominance of proprietary, closed source software model," and thus this novel term would be needed. The article describes the need to balance profit motives with those of acceptable open community values. As listed in the article, the characteristics of this change were to include (compared to earlier open source) purposive strategies of commercial actors related to process planning, process stages of analyses and design spread to vertical domains, less bazaar-like development models, increasingly paid development effort, more visible product applications, novel business strategies, better product support, and an increased number of potential licenses (Fitzgerald, 2006).

From OSS 2.0 to Blockchains (2006–2020)

Over time, there has been a steady increase in the industrial use of OSS in open developer communities on the Internet (Fitzgerald, 2006), as well as OSS-like practices for software production in organizations (Van Der Linden et al., 2009).

In 2008, a whitepaper published in an Internet forum under the pseudonym Satoshi Nakamoto described the first implementation of blockchain technology: bitcoin cryptocurrency (Nakamoto, 2008). This whitepaper provided a roadmap to build a decentralized payment system and a store of value. The decentralized nature, shared state of the blockchain and append-only data structure guarantee a reliable record of earlier transactions—a strong requirement for any payment system, but a usable concept for a number of other potential uses.

Bitcoin also relied on public key infrastructure as an addressing and security system for the payments and a method called proof-of-work as a security mechanism. The miners maintaining the network were granted cryptocurrency—this mechanism also controlled the supply of money and initially incentivized the individual miners.

Bitcoin sustained the negative publicity on energy use, compromised wallets and exchanges and obscure personalities, not entirely unlike OSS 10 years before. Therefore, a similar move to OSS occurred: Bitcoin's underlying infrastructure was decoupled from any particular cryptocurrency and renamed as a blockchain following the logic of transactions that were stored in a decentralized way—as a chain of blocks.

Bitcoin Foundation was founded to support the development efforts that relied on the global OSS developer community (Teigland et al., 2013). Maybe the second best-known cryptocurrency, Ethereum was introduced in 2013. A main driver for its initial development initiatives was the disagreements concerning what kinds of scripting to include in Bitcoin to enable smart contracts—these debates later resulted in Ethereum, which quickly attracted a large following. The issues around scripting language were important, as they enabled the implementation of smart contracts on Ethereum (Szabo, 1996).

At this point, OSS had become so mainstream that initial Bitcoin and Ethereum quickly applied OSS licensing and practices to their own work[6]. This meant that OSS research should be directly applicable to these best-known public blockchain projects.

DISCUSSION

Thus, OSS research seems to offer several insights that may be usable in the blockchain context regarding how to solve different kinds of tension in voluntary communities and contexts where commercial companies also are involved.

Openness of the code is an interesting point of departure, but more critical questions may be related to guaranteeing the different actors' incentives and matching divergent interests.

This paper investigated three ways OSS research can help to explain blockchain-related phenomena. The discussion is divided into three areas: (1) blockchain and OSS 2.0, (2) community development and governance, and (3) forks.

Blockchain and OSS 2.0

For-profit actors traditionally leveraged OSS using "generic" OSS business models (Hecker, 1999; Osterwalder et al., 2005), hybrid models (Sharma et al., 2002) or management strategies (West, 2003). Some suggested ways for companies to approach OSS included using OSS tools, for example CASE tools (Toth, 2006); integrating OSS licensed systems into existing systems (Li et al., 2009); companies participating in OSS development in the open (Dueñas et al., 2007) or even providing company products or services as openly licensed (Frost, 2007; Santos, 2008).

An early discussion was about how to build a mutually beneficial relationship between commercial organizations and communities (Dahlander and Magnusson, 2008). These relationships included (1) access (resource base via communities), (2) aligned company strategies with the chosen communities, and (3) assimilation of communities to integrate and share results.

Where OSS 2.0 had grown to rely on commercial involvement, enthusiasm around Bitcoin/Ethereum from companies, entrepreneurs, and industries quickly grew in numbers compared to other early OSS projects.

Another interesting similarity is the originally strong political polarization (especially free software and Bitcoin) and hyperbole (OSS and blockchain as revolutionary change) around the innovations. Many early enthusiasts in OSS and blockchain were obviously motivated by the "bazaar-style" decentralizations that both technologies offered. Coining the terms OSS 2.0 and blockchain can be seen as similar rhetorical moves and efforts to bridge divergent interests of for-profit organizations and developer communities working in the same area.

Community Development and Governance

Early literature reviews showed wide proliferation of the OSS term in different streams of literature in software engineering, computer science, information systems,

[6] Both projects use a well-known OSI-approved license (MIT license).

psychology, law, and others (Gacek and Arief, 2004; Von Krogh and Von Hippel, 2006). There was also plenty of heterogeneity in how the different disciplines approached the phenomenon in terms of epistemology, theory and methods.

OSS research can be characterized in several ways. Von Krogh and Von Hippel (2006) offered one early classification of the relevant literature by dividing the research into three distinct groups of studies: (1) motivations for OSS contributions (Hars and Ou, 2002; Ke and Ping, 2010; for example, incentives to contribute); (2) OSS governance, organization and the innovation process (Crowston and Howison, 2006; for example, structures and communication); and (3) competitive dynamics (Bonaccorsi et al., 2006; for example, licensing, revenue and business models).

OSS governance means project direction, control, and coordination (Markus, 2007). Interesting and still relevant questions were raised, such as what are the different motivations for the individual developers, how to build gatekeeping processes and protect the codebase, how to organize the production of software into tasks that can be voluntarily assigned, how to develop thriving community cultures and practices, and how to work with commercial actors. This literature can be divided into literature streams: (1) incentives for independent developers to participate in open efforts and software development (for example, Lerner and Tirole, 2002), (2) efforts to provide support for the coordination activities (for example, Crowston et al., 2005), and (3) encouragement to build a welcoming culture in a community. There is a rich literature in economics concerning governance questions related to open source and open innovation in general. A key stream of governance discusses OSS in relation to innovation or as a novel type of commons: knowledge commons (Frischmann et al., 2014) or innovation commons (Allen and Potts, 2016; Potts, 2019).

Over time, many OSS projects developed different kinds of formal and informal governance structures, for example to handle their rights, IPRs and revenues. A more formal stewardship vehicle was the formation of different kinds of foundations (Lindman et al., 2017).

In many blockchain projects, the core developers and adjoining community are heavily invested in the technology (i.e., they usually hold plenty of currency). The impact of this on organizational dynamics is unclear, but it is easy to speculate that this would be a different factor compared to earlier OSS development efforts.

Forking

The reemergence of forks in blockchain[7] also offer opportunities for OSS research. Forking has a long tradition in software engineering[8]. However, the term's meaning has not been clear-cut, even related to OSS, but usually fork meant a situation in OSS where the developer community disagrees on the development roadmap (or a different focal issue), which results in a situation where several different competing and backward-incompatible versions of the code base are in use.

For OSS, forks are seen as a safeguard of openness and as severely detrimental to the development efforts because forks dilute the (already scarce) contributions and developers among the different versions (Fogel, 2006: 88, Moody, 2002). Viseur (2012) found that the majority of forks studied were motivated by a need for technical specialization, and forks rarely were followed by the extinction of the original. Based on an in-depth analysis of 220 forks, Robles and González-Barahona (2012) claimed forks took place in all software domains, could be friendly or competitive and had become more frequent in recent years. One important development in this space was that most OSS projects moved their revision control to GitHub—a company and service that had its own vocabulary: for example, any novel development (version) effort from the same codebase was referred to as "a fork."

For blockchain, Bitcoin and Ethereum have undergone several so-called hard forks (backward-incompatible) and competitive forks, thus this research is becoming pivotal again[9]

However, it is noteworthy that, where traditional OSS forks were relatively rare (because of their detrimental impact), blockchains are suffering from them constantly. Another difference is related to the speculative part of a hard fork. Traditional OSS users could select the fork they preferred to and that was it. However, for cryptocurrency blockchains, the backwards-incompatible fork had a direct impact on the blockchain network membership and even the value of the specific coins (incumbent and fork). The coin ecosystem is thus connected to other users (an unconnected payment system would not make much sense), which also means that the protocols (versions) of the cryptocurrency software need to be compatible. Therefore, users need to agree on incompatible protocol revisions (version upgrades). For traditional software updates, individual users could make selections on the software version they chose, depending, for example, on how mature the version was or whether it had some functionality they required, which would have minimal impact on any of the other users.

This difference and direct relation to coin value seems to generate radically different incentives for evaluating the project roadmap, governance and impacts of potential forks for developers. However, this initially could mean less, not more, forks. Thus, the current situation of increasing forks is theoretically interesting.

[7] For summaries of OSS fork discussions, see Nyman (2015) and Robles and González-Barahona (2012).

[8] See Nyman and Laakso (2016) for a historical account where the word originates in computing.

[9] Here is an indicative list of Bitcoin forks (105 forks at the time of writing) https://forkdrop.io/list-of-bitcoin-forks For Ethereum, here is one recent list of the most influential forks: https://medium.com/mycrypto/the-history-of-ethereum-hard-forks-6a6dae76d56f

CONCLUSION

This section presents some implications for the practitioners working in the area, lists the limitations and potential avenues for studies and concludes the article.

Managerial Implications

The literatures offer several managerial implications.

- Heated and philosophical discussions around "what constitutes a blockchain" are not useful for disseminating a specific technology. OSS teaches us that umbrella terming can be a successful strategy to gain credibility or "winning" (i.e., gaining usage) and does not automatically disqualify the term from more philosophical considerations, as this worked for "OSS" and "OSS 2.0." Some of the early noncommercial and anti-commercial activity related to Bitcoin were probably not helpful in terms of credibility. However, it is important to note that this does not mean dampening necessary critiques and concerns around technology. For blockchain, it is important to address the expectations set, which are a bit different for OSS. OSS did not really suffer from the overblown expectations and hype problems blockchain did. If anything, OSS had a continuous credibility problem, a handicap for mainstream usage, and had to prove the individual developers' "business models" and "incentives."
- Aligning incentives and building different kinds of rapport to OSS communities are similar to blockchain and other OSSs. Evaluating different OSS projects and communities to identify how critical they are for future company road plans is similar to blockchain projects and other projects—as does the need for companies to be aware of what is happening in communities by formal means (appointing ambassadors) or informal means (letting developers work on relevant OSS projects).
- Literature on the role of different kinds of relevant foundations and more formal governance mechanisms seems immediately applicable to the public blockchain space. Bitcoin and Ethereum have experience on these forms, but they are likely important for other smaller novel projects too. Other, more informal governance mechanisms also seem relevant for blockchain practitioners.
- The earlier literature on forking seems directly applicable, despite the novel challenges in the cryptocurrency area: hard forks in this area generate de facto novel cryptocurrency. Even though the bigger blockchain developer communities have a larger number of developers than many traditional OSS examples, from the software development perspective, forks always carry risks related to diluting development efforts. However, this issue seems complicated by how large disagreements or confusion over prioritization and roadmap improvements also lead to a drastic slowing of development. Thus, the fork may be able to expedite developments in such situations.

Limitations and Avenues for Future Research

This paper comes with several different limitations that call for further research efforts in the area. First, there are some immediate differences between the selected public blockchain projects, other blockchain projects and earlier OSS projects. Second, the two selected blockchain projects have unique characteristics that differentiate themselves from other blockchains. Third, Bitcoin and Ethereum ecosystems are separate from earlier OSS projects, requiring further study around these areas.

Many of the early OSS projects were relatively small in terms of developers—if the 10 most popular projects were left aside, most of the other OSS projects had one or two core developers. Code openness also invited potential sporadic contributions, but those often went through some of the core developers who maintained gatekeeper roles in the projects. However, this did not mean there would be only one or two persons on the entire development effort, even though contributions might have come from a small group of known developers. That said, many of the projects were relatively small—meaning that organizational dynamics revolved around one or few people.

This paper only focuses on literature and ecosystems on two public blockchains: Bitcoin and Ethereum. Both of these chosen projects have evolved and forked into continuing projects several times. These two blockchains should not be seen as generally representative of all blockchain technologies or blockchain research theories. Both are also closer to traditional OSS communities than some other non-OSS licensed blockchain communities and projects are. Company-driven private and permissioned blockchains likely have different dynamics and government.

Tokenization of governance is an interesting future direction. What changes when moving from voluntary-based task assignment to tokenized incentive? One intriguing aspect is the impact to the production process and outcome software quality.

Thus, more theorizing and empirical research are needed around the suggested themes and this paper should stimulate further activities in this space.

DATA AVAILABILITY STATEMENT

The dataset presented in this study can be found in online repositories. The name of the repository and accession number can be found in the article.

AUTHOR CONTRIBUTIONS

The author confirms being the sole contributor of this work and has approved it for publication.

REFERENCES

Allen, D., and Potts, J. (2016). "The origin of the entrepreneur and the role of the innovation commons," in *Conference Paper for 86th Annual Meeting of the Southern Economic Association Conference* (Washington, D.C).

Beck, R., Avital, M., Rossi, M., and Thatcher, J. B. (2017). Blockchain technology in business and information systems research. *Bus. Inf. Syst. Eng.* 59, 2017, 381–384. doi: 10.1007/s12599-017-0505-1

Benkler, Y. (2006). *The Wealth of Networks*. London: Yale University Press.

Bonaccorsi, A., Giannangeli, S., and Rossi, C. (2006). Entry strategies under competing standards: hybrid business models in the open source software industry. *Manag. Sci.* 52, 1085–1098. doi: 10.1287/mnsc.1060.0547

Crowston, K., and Howison, J. (2006). Hierarchy and centralization in free and open source software team communications. *Knowl. Technol. Policy* 18, 65–85. doi: 10.1007/s12130-006-1004-8

Crowston, K., Wei, K., Li, Q., Eseryel, U. Y., and Howison, J. (2005). "Coordination of free/libre and open source software development," in *Proceedings of the International Conference on Information Systems (ICIS 2005)* (Las Vegas, NV).

Dahlander, L., and Magnusson, M. (2008). How do firms make use of open source communities? *Long Range Plann.* 41, 629–649. doi: 10.1016/j.lrp.2008.09.003

Dinkelacker, J., Garg, P., Miller, R., and Nelson, D. (2002). "Progressive open source," in Proceedings of the 24th International Conference on Software Engineering. ICSE 2002 (Orlando, FL).

Dueñas, J., Parada, H., Cuadrado, F., Santillán, M., and Ruiz, J. (2007). Apache and eclipse: comparing open source project incubators. *IEEE Softw.* 24, 90–98. doi: 10.1109/MS.2007.157

Fitzgerald, B. (2006). The transformation of open source software. *MIS Q.* 30, 587–598. doi: 10.2307/25148740

Fogel, K. (2006). Producing Open Source Software: How to Run a Successful Free Software Project. Newton, MA: O'Reilly.

Frischmann, B., Madison, M., and Stranburg, K. (2014). "Introduction and Chapter 1," in *Governing Knowledge Commons*, eds B. Frischmann, M. Madison, and J. Katherine Strandburg (Oxford: Oxford University Press), 1–44.

Frost, R. (2007). Jazz and the eclipse way of collaboration. *IEEE Softw.* 24, 114–117. doi: 10.1109/MS.2007.170

Gacek, C., and Arief, B. (2004). The many meanings of open source. *IEEE Softw.* 21, 34–40. doi: 10.1109/MS.2004.1259206

Hars, A., and Ou, S. (2002). Working for free? motivations for participating in open-source projects. *Int. J. Electron. Commer.* 6, 25–39. doi: 10.1080/10864415.2002.11044241

Hauge, Ø., Ayala, C., and Conradi, R. (2010). Adoption of open source software in software-intensive industry–a systematic literature review. *Inf. Softw. Technol.* 52, 1133–1154. doi: 10.1016/j.infsof.2010.05.008

Hecker, F. (1999). Setting up shop: the business of open-source software. *IEEE Softw.* 16, 45–51, doi: 10.1109/52.744568

Ke, W., and Ping, Z. (2010). Extrinsic motivations in open source software development efforts and the moderating effects of satisfaction of needs. *J. Assoc. Inf. Syst.* 11, 784–808. doi: 10.17705/1jais.00251

Lerner, J., and Tirole, J. (2002). Some simple economics of open source. *J. Ind. Econ.* 50, 197–234. doi: 10.1111/1467-6451.00174

Li, J., Conradi, R., Bunse, C., Torchiano, M., Slyngstad, O. P. N., and Morisio, M. (2009). Development with off-the-shelf components: 10 facts. *IEEE Softw.* 26, 80–87. doi: 10.1109/MS.2009.33

Lindman, J., Berryhill, J., Welby, B., and Barbeiri, M. P. (2020). *The Uncertain Promise of Blockchain for Government, OECD Working Papers on Public Governance, No. 43*. Paris: OECD Publishing.

Lindman, J., Rossi, M., and Tuunainen, V. (2017). "Opportunities and risks of Blockchain Technologies in payments–a research agenda," in *Proceedings of Hawaiian International Conference of Systems Sciences (HICSS 2017)* (Waikoloa Village: Hawaii).

Markus, M. L. (2007). The governance of free/open source software projects: monolithic, multidimensional, or configurational? *J. Manag. Govern.* 11, 151–163. doi: 10.1007/s10997-007-9021-x

Marsan, J., Carillo, K., and Negoita, B. (2020). Entrepreneurial actions and the legitimation of free/open source software services. *J. Inf. Technol.* 35, 143–160. doi: 10.1177/0268396219886879

Moody, G. (2002). *Rebel Code: Linux and the Open Source Revolution*. New York, NY: Basic Books.

Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system*.

Nyman, L. (2015). *Understanding Code Forking in Open Source Software: An Examination of Code Forking, Its Effect on Open Source Software, and How it Is Viewed and Practiced by Developers*. Helsinki: Hanken School of Economics.

Nyman, L., and Laakso, M. (2016). Notes on the history of fork-and-join. *IEEE Ann. Hist. Comput.* 38, 84–87. doi: 10.1109/MAHC.2016.34

Osterwalder, A., Pigneur, Y., and Tucci, C. L. (2005). Clarifying business models: Origins, present, and future of the concept. *Commun. Assoc. Inf. Syst.* 16, 1–25. doi: 10.17705/1CAIS.01601

Perens, B. (1999). "The open source definition," in *Open Sources: Voices from the Open Source Revolution*, eds C. Dibona and S. Ockman (Sebastopol, CA: O'Reilly Media), 171–188.

Potts, J. (2019). *Innovation Commons: The Origin of Economic Growth*. New York, NY: Oxford University Press.

Raymond, E. (1999). *The Cathedral and the Bazaar–Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA: O'Reilly Associates.

Robles, G., and González-Barahona, J. (2012). "A comprehensive study of software forks: dates, reasons, and outcomes," in *Proceedings of the 8th IFIP WG 2.13 International Conference, OSS 2012* (Hammamet).

Santos, C. (2008). Understanding partnerships between corporations and the open source community: a research gap. *IEEE Softw.* 25, 96–97. doi: 10.1109/MS.2008.167

Schlagwein, D., Conboy, K., Feller, J., Leimeister, J. M., and Morgan, L. (2017). Openness" with and without information technology: a framework and a brief history. *J. Inf. Technol.* 32, 297–305. doi: 10.1057/s41265-017-0049-3

Sharma, S., Sugumaran, V., and Rajagopalan, B. (2002). A framework for creating hybrid-open source software communities. *Inf. Syst. J.* 12, 7–25. doi: 10.1046/j.1365-2575.2002.00116.x

Stallman, R. (2002). *Free Software, Free Society: Selected Essays of Richard M.* Boston, MA: Stallman. Free Software Foundation.

Swanson, B., and Ramiller, N. (1997). The organizing vision in information systems innovation. *Organ. Sci.* 8, 458–474. doi: 10.1287/orsc.8.5.458

Szabo, N. (1996). "Smart contracts: building blocks for digital markets," in *EXTROPY: The Journal of Transhumanist Thought tei*, ed M. More and T. O. Morrow (Riverside, CA: Extropy Institute).

Szczepanska, A., Bergquist, M., and Ljunberg, J. (2005). "High noon at OS corral: duels and shoot outs in open source discourse," in *Perspectives on Free and Open Source Software*, eds J. Feller, B. Fitzgerald., S. A. Hissam, and K. R. Lakhani (Cambridge, MA: MIT Press), 431–446.

Teigland, R. Yetis, Z., and Larsson, T. (2013). *Breaking Out of the Bank in Europe - Exploring Collective Emergent Institutional Entrepreneurship Through Bitcoin (May 11, 2013)*. Available online at: https://ssrn.com/abstract=2263707

Toth, K. (2006). Experiences with open source software engineering tools. *IEEE Softw.* 23, 44–52. doi: 10.1109/MS.2006.158

Van Der Linden, F., Lundell, B., and Marttiin, P. (2009). Commodification of industrial software: a case for open source. *IEEE Softw.* 26, 77–83. doi: 10.1109/MS.2009.88

Viseur, R. (2012). Forks impacts and motivations in free and open source projects. *Int. J. Adv. Comput. Sci. Appl.* 3, 1–6. doi: 10.14569/IJACSA.2012.030221

Von Krogh, G., and Von Hippel, E. (2006). The promise of research on open source software. *Management Sci.* 52, 975–983. doi: 10.1287/mnsc.1060.0560

Weber, S. (2004). *The Success of Open Source*. Harvard, MA: Harvard University Press.

West, J. (2003). How open is open enough? melding proprietary and open source platform strategies. *Res. Policy* 32, 1259–1285. doi: 10.1016/S0048-7333(03)00052-0

Xu, L., and Brinkkemper, S. (2007). Concepts of product software. *Eur. J. Inf. Syst.* 16, 531–541. doi: 10.1057/palgrave.ejis.3000703

Xu, X., Weber, I., Staples, M., Zhu, L., Bosch, J., et al. (2017). "A taxonomy of blockchain-based systems for architecture design," in *Software Architecture (ICSA), 2017 IEEE International Conference on (IEEE, 2017)* (Gothenburg), 243–252.

Conflict of Interest: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be constructed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.