Open Source Operational Risk: Should Public Blockchains Serve as Financial Market Infrastructures?

Angela Walch

chapter to be included in
Handbook of Digital Banking & Internet Finance, Vol. 2
eds. David Lee Kuo Chuen & Robert H. Deng
Elsevier, Forthcoming 2017

Every reader of this *Handbook* will be well aware that blockchain technology, also called 'distributed ledger technology' or 'DLT', is all the rage in financial circles at the moment. One cannot escape white papers by various banks and consulting firms, speeches by central bankers, and a deluge of articles, books, conferences, summits, and workshops proclaiming the imminent transformation of the financial system by this revolutionary technology. The Gardner Hype Cycle put blockchain technology at virtually the top of its hype cycle curve in the summer of 2016, indicating that the zeitgeist around blockchain technology is soon to fall into the "trough of disillusionment" as sky-high expectations collide with harsh realities (Burton, B. and D. Willis 2016).

This chapter delves into some of those harsh realities, as it focuses on the risks created by the use of what I call 'grassroots' open source software[1] methods in the operation of public blockchains, and the resulting fragility of any systems that rely on public blockchains as underlying technological infrastructure. Public blockchains, otherwise referred to as 'open' or 'permissionless' blockchains, allow anyone to become part of the computer network that maintains the blockchain; to join, one simply downloads and begins to run the applicable software. Private blockchains, otherwise referred to as 'closed' or 'permissioned' blockchains, allow only those who have received 'permission' to join the computer network that maintains the blockchain, thus limiting the transaction processing network to those who are known and trusted. Public and private blockchains are diametrically opposed to one another, and the seemingly simple decision about access to the network of transaction processors fundamentally changes the risk profile (as well as the capabilities and emergent properties) of a blockchain.

This chapter limits its analysis to public blockchains, and explores how the use of three common practices from the grassroots open source software world gives rise to operational risks for these blockchains. These practices are: (1) the use of the informal, semi-decentralized grassroots open source software development process to maintain the blockchain software; (2) the use of the funding model (or lack thereof) for grassroots open source software development; and (3) the practice of forking software code that is an inherent feature of open source software.

[1] In this chapter, I distinguish between "grassroots" open source software ("community-developed," Nyman 2015, p. 24) and "corporate" open source software. The distinction between the two is generally that "grassroots" open source software emerges organically from and is maintained by a community of software developers (sometimes with the assistance of a purpose-built non-profit foundation), while a "corporate" open source software project is created, owned, and controlled by a formal business entity, with some sort of participation from the larger developer community (Nyman 2015).

As each of these practices generates operational risks for public blockchains, systems built on these structures (such as financial market infrastructures, or "FMIs") would be similarly subject to these systemic vulnerabilities. Threats to financial market infrastructures are threats to broader financial stability, which makes the financial sector's fascination with all things 'blockchain' extremely significant. While existing FMIs are of course subject to systemic risks, it is important not to gloss over serious operational risks in certain forms of blockchain technology in the rush to fix our much-maligned existing FMIs. Important tradeoffs exist in repairing or replacing our current financial market infrastructures, and it is vital to assess those tradeoffs through crystal-clear, rather than rose-tinted, lenses.

In Part I of this chapter, I describe the excitement about blockchain technology in the financial sector, as well as the features of the technology that are seen as attractive and transformational. Blockchain technology appears to offer the silver bullet of solutions to financial practices in providing reliability and certainty.

In Part II, I provide background information on how financial market infrastructures are treated by global financial regulators. Given FMIs' acknowledged significance to maintaining global financial stability, their reliability and resilience are crucial, and I note particularly how regulators have identified governance and operational risks as critical ones for FMIs to manage.

In Part III, I provide the core contribution of this chapter, in analyzing three practices common to grassroots open source software development that create systemic instability for public blockchains: (1) informal governance of the software code; (2) funding software development and maintenance in an uncertain or experimental way (if at all); and (3) the practice of forking the software code to make changes to it. I explicate how each of these practices produces significant operational risks for public blockchains, as already demonstrated by real-world events with both Bitcoin and Ethereum, the best-known public blockchains.

In Part IV, I reflect on the implications of these operational risks in how we choose to use public blockchains, as well as their implications for the use of grassroots open source software practices in critical systems, generally. As some think that blockchain technology will revolutionize virtually every system of record-keeping or exchange that we have, it is important to reexamine the basic attributes of the technology to understand the tradeoffs we would make in choosing to integrate public blockchains widely.

I. FINANCIAL SECTOR HYPE

The financial world has become obsessed with blockchain technology. There are infinite conferences and workshops devoted to it, blockchain and fintech thought leaders ply their wisdom through Twitter and other forms of media, and every significant financial player, from J.P. Morgan to DTCC all the way up to the world's central banks, is experimenting with the technology, and proclaiming that it will transform the financial sector. Mark Carney, Governor of the Bank of England, discussed the technology in his important Mansion House speech in June 2016 (Carney 2016), and blockchain technology proponents presented to representatives from 90 central banks at the Federal Reserve in June 2016 (Rapier 2016). Blockchain technology is the ultimate trend at the moment, and no one wants to be left behind.

In this Part, I describe the financial sector's great interest in blockchain technology, the features of the technology that are celebrated by the sector as transformative, and the benefits that the sector hopes to achieve through the use of the technology.

Once this foundation is laid, I can demonstrate more clearly with my analysis in Part III how common practices from grassroots open source software development make public blockchains, as currently structured, inappropriate for the financial sector's plans.

Who is Interested in Blockchain Technology?

Since the summer of 2015, the financial world has been intoxicated by blockchain technology. Around 70 banks have joined together in a consortium called R3Cev to develop distributed ledger technology together (Eha 2016). Hyperledger, an open source consortium for the development of common blockchain tools, was formed in conjunction with the Linux Foundation, and has received code contributions from Digital Asset Holdings and IBM, among others (Rizzo 2016). The Bank of England has formed a partnership with Big-Four Accounting firm PWC to investigate the use of distributed ledgers in the financial sector (PWC 2016). Numerous central banks, financial regulators, and international economic organizations have spoken out about the promise of blockchain technology to revolutionize financial systems. A belief in the benefits of blockchain technology to the financial sector is clearly widely shared, and a plethora of blockchain-related books has been released in the past year, proclaiming the technology's virtues. It is fair to say that the consensus view is that blockchain technology is a massive innovation that will transform and improve financial structures and practices. Dissenting voices are few and far between.[2]

What Do They Like About It?

Blockchain technology is attractive to the financial sector due to its most celebrated attributes. In report after report from central banks, prestigious consulting and financial institutions, and international economic groups, the following descriptors are repeatedly cited as potentially transformative for finance:

- *Immutability*
- *Trustlessness*
- *Visibility / Transparency*
- *Resilience*

What all of these features have in common is that they suggest that the technology is something reliable – that it can be counted on, whether that reliability has to do with the truth of what the blockchain displays or its continued operation in a crisis. In this section, I discuss in more detail what the financial world likes about these features, and industry visions of how these features would improve it.

[2] Izabella Kaminska of the *Financial Times*, Matt Levine of *Bloomberg View*, and Steve Wilson of Constellation Research have been among the few prominent critics of the hype surrounding blockchain technology.

Immutability: The ledgers that operate through blockchain technology are said to be immutable – i.e., unchangeable. This means that once an entry is added to the blockchain, it cannot be altered or removed. This is attractive to participants in the financial sector because it means they can rely on the truth of the ledger – immediately, and without having to expect corrections to it. An important implication of this is that amounts set back in reserve to address settlement risk could be foregone, and finance could proceed more efficiently. (As DTCC's 2016 White Paper on blockchain technology noted, however, immutability is not particularly attractive to financial transactions, as there are inevitably errors and fraud in the real world that require corrections to be made (DTCC 2016, p. 8). The prominent consulting firm Accenture recently announced that it is patenting an "editable blockchain," triggering widespread derision from public blockchain advocates (Arnold 2016).)

Trustlessness: The decentralized nature of blockchain technology is attractive because it eliminates the need to count on a central trusted party to operate it – i.e., to make updates to the ledger and to keep the technology running. Thus, you no longer need the intermediaries who serve as aggregators of risk – known as central counterparties. You trust that the transaction processing network—through the magic of cryptography, algorithms, and a defined process for achieving consensus—is maintaining a truthful ledger.

This is a very attractive idea. In the finance world, trust is everything, because all the outcomes are determined by how trustworthy a counterparty is. Will it pay you back as promised? Will the company do as well as it predicted? All of finance is essentially a gamble on the trustworthiness of the other side, and being able to eliminate even one uncertainty (the trustworthiness of a central counterparty) is extremely valuable. (For now, we will gloss over the fact that, even with a blockchain, one must trust the integrity of the code, the developers, and the transaction processors. The rosy view is that eliminating trust in a central operator of the ledger is an unsullied innovation.)

Visibility/Transparency: In traditional public blockchains like Bitcoin or Ethereum, the common ledger that is the blockchain is visible to all nodes (parties) in the network. So everyone sees the same thing. This means that it is possible to evaluate risk and therefore price it more accurately. If risk decisions are based on current and reliable information, the theory goes, they will be better decisions. Blockchain or distributed ledger technology allows for this visibility and transparency because the ledger is distributed to multiple parties simultaneously. This means that every participant in the blockchain or distributed ledger network has a live, up-to-date copy of the ledger, and sees any changes in real time as they are made. Everyone sees the same thing, and knows that the ledger represents truth. This is a boon to the financial sector because time delays in confirming trades add uncertainty to the process, and force participants to reserve resources against this settlement risk. With instantaneous settlement, it is said, there is no need to reserve against settlement risk, saving lots of money.

Some have gone so far as to say that blockchain technology could have prevented the Financial Crisis, particularly the fall of Lehman Brothers, as the Department of the Treasury and the Federal Reserve would have had real-time information to help them determine whether Lehman was solvent and could be saved (Giancarlo 2016). In a world that is looking for ways to

avoid crises while maintaining a growing economy, blockchain technology appears to offer potent tools well-suited for this purpose.

Resilience: The critical importance of financial market infrastructures has long been recognized, but has received renewed attention following the 2008 Financial Crisis. These pathways of communication can rapidly transmit financial crises, and their failure can trigger or worsen a crisis. Global financial regulators therefore set standards for financial market infrastructures, such as payment, clearing, and settlement systems. In the past few years, cyber-resilience of financial market infrastructures has been particularly emphasized (Bank for International Settlements 2015), as hack after hack has hit major companies, government agencies, and even the SWIFT consortium operated by the world's financial institutions (Perlroth and Corkery 2016). A systemic outage in a financial market infrastructure is a worst case scenario for the financial system, so blockchain technology's reputation for resilience is highly attractive to global financial regulators.

Blockchain technology's purported resilience derives from its decentralized structure. As the network operates on a peer-to-peer basis rather than through a central server, there is no single point of failure in the network. Theoretically, there are as many up-to-date and operational copies of the distributed ledger as there are nodes in the network. This is highly desirable for a digital infrastructure, as it would be extraordinarily difficult to knock out all nodes simultaneously. Central banks and other international economic organizations have commented on this as an extremely attractive feature of blockchain technology (See, e.g., Carney 2016).

* * *

All of these features – immutability, trustlessness, visibility/transparency, and resilience – add up to increased reliability. Given that a major function of the financial sector is simply keeping track of who has what, a reliable record-keeping system is vital. If blockchain technology offers a more reliable record-keeping system, then risk is reduced because something (truth, timeliness, long-term existence and ongoing operation) can be counted on. This means that more risk can potentially be taken in other areas, particularly if the whole concept of settlement risk evaporates because settlements are instantaneous. If the financial sector can know things in real time and those things can't be changed, then people within the sector can make decisions more quickly, with the confidence that they are standing on something certain, rather than upon shifting sands. And, these factors all should lead to big savings for the financial sector, by eliminating steps (and people) from (as well as speeding up) all sorts of processes. Increased savings and greater efficiencies should yield bigger profits, making the promise offered by this technology heady indeed.

Thus, the finance industry contemplates using blockchain technology in virtually every trading, settlement, clearing, and recording function that it engages in, from stocks, to bonds, to foreign exchange, to derivatives. If it is traded or recorded in the financial sector, blockchain technology, proponents say, is going to revolutionize it.

II. FMIs and Operational Risk

Before delving into the operational risks stemming from the use of grassroots open source software practices in public blockchains, this Part II provides a brief overview of what financial market infrastructure is, its extreme importance to global financial stability, and how global regulators treat operational risk in existing financial market infrastructures.

Financial market infrastructures are "multilateral systems among participating financial institutions…used for the purposes of clearing, settling, or recording payments, securities, derivatives, or other financial transactions," which "include payment systems, central securities depositories, securities settlement systems, central counterparties, and trade repositories" (Federal Reserve 2016, p. 3). These systems allow our vast economies to keep track of who owns (and owes) what.

Unsurprisingly, the uninterrupted operation of financial market infrastructures is extraordinarily important to global financial stability. Failure in a system that functions as financial market infrastructure could disrupt financial markets and affect the public's faith in the financial system (Federal Reserve 2016).

Given the massive problems that could be caused by failures of financial market infrastructures, regulators around the world have worked together to adopt principles to help FMIs mitigate their risks. The goal behind these principles is generally "to foster the safety and efficiency of payment, clearing, settlement, and recording systems and to promote financial stability, more broadly" (Federal Reserve 2016, p. 3). Many countries have based their guidance to FMIs on the April 2012 *Principles for Financial Market Infrastructures* (PFMI) report by the Bank for International Settlement's Committee on Payment and Settlement Systems (CPSS)[3] and Technical Committee of the International Organization of Securities Commissions (IOSCO).

The risks that the international guidelines for FMIs are intended to address include credit risk, operational risk, liquidity risk, legal risk, systemic risk, general business risk, and custody and investment risk (Federal Reserve 2016; *PFMI* 2012). In this Chapter, I focus on operational risk, as it is important to understand whether public blockchains can live up to their reputation for reliability, so prized by the financial sector. The Federal Reserve's definition of operational risk is fairly standard:

> The risk that deficiencies in information systems or internal processes, human errors, management failures, or disruptions from external events will result in the reduction, deterioration, or break-down of services provided by the [financial market infrastructure]…includ[ing] physical threats, such as natural disasters and terrorist attacks, and information security threats, such as cyberattacks. Further, deficiencies in information systems or internal processes include errors or delays in processing, system outages, insufficient capacity, fraud, data loss, and leakage (Federal Reserve 2016, p. 5).

Essentially, operational risk is a catch-all sort of risk that deals with unexpected external events, and problems caused by human imperfections. These are precisely the types of risks that, in Part

[3] Since September 2014, the CPSS has been known as the Committee on Payments and Market Infrastructures.

III, I argue are generated by the use of common grassroots open source software practices in public blockchains.

To help mitigate operational risks, the *Principles for Financial Market Infrastructures* include these standards:

> *Principle 2: Governance:* An FMI should have governance arrangements that are clear and transparent, promote the safety and efficiency of the FMI, and support the stability of the broader financial system, other relevant public interest considerations, and the objectives of relevant stakeholders.[4]
>
> *Principle 3: Framework for the comprehensive management of risks:* An FMI should have a sound risk-management framework for comprehensively managing legal, credit, liquidity, operational, and other risks.
>
> *Principle 17: Operational risk:* An FMI should identify the plausible sources of operational risk, both internal and external, and mitigate their impact through the use of appropriate systems, policies, procedures, and controls. Systems should be designed to have a high degree of security and operational reliability and should have adequate, scalable capacity. Business continuity management should aim for timely recovery of operations and fulfillment of the FMI's obligations, including in the event of a wide-scale or major disruption (*PFMI* 2012, pp. 1-3).

As I will discuss in Part III, these widely-recognized principles for mitigating operational risk in FMIs are fundamentally at odds with common practices of public blockchains, and it is very difficult to see how they could be met in public blockchains without altering certain grassroots open source software practices that help contribute to their openness. Thus, if we were to build financial market infrastructures atop public blockchains as they exist currently, we would be accepting a new flavor of operational risk. The question becomes, then, whether the benefits of public blockchains serving as the underlying technology of FMIs (e.g., reduction of settlement risk) can justify the acceptance of this type (and potentially higher level) of operational risk.

III. Open source Operational Risks of Public Blockchains

Reliable, certain, resilient (yet speedy) systems are the Holy Grail for the financial world, and many believe that blockchain technology represents the finding of the Grail for the record-keeping systems that comprise much of finance (Kaminska 2016). It remains to be seen, however, whether this quest has been completed.

An important decision remains to be made as blockchain development proceeds apace: whether public or private blockchains will be used to transform the world of finance. Most experimentation and development work on blockchain technology in the financial sector has involved private, or closed, blockchains. However, the debate over the appropriate form of

[4] I include Principle 2 regarding governance because poor governance can generate the human problems that are part of the wider concept of operational risk.

blockchain technology (public or private) has not yet been resolved. A number of prominent, respected players in the blockchain and finance worlds have noted that public blockchains like Bitcoin and Ethereum remain in the hunt. For instance, MUFG, the world's third largest bank, announced in October 2016 that it was working with Coinbase, a Bitcoin exchange, to conduct cross-border payments through Bitcoin (Eha 2016). And SEC Chair Mary Jo White stated in November 2016 that the SEC is looking at whether blockchain technology used in the financial section will be permissioned (White 2016), suggesting that the debate between public and private blockchains has not yet ended.

This paper contributes to the public-versus-private blockchain debate, explicating how the use of traditional grassroots open source software practices in public blockchains would expose any financial market infrastructures they undergird to new and potentially increased operational risks in exchange for the benefits they seductively promise. There are tradeoffs to all improvements we make, and in this case, the new operational risks seem quite hefty.

In this Part III, I explore a set of operational risks generated by the use of customary grassroots open source software practices in the creation and maintenance of public blockchains. The risks and practices that I examine include:

1) the risk of impeded decision-making about changes to the software code, resulting from using the typical informal grassroots open source software development process;

2) the risk that the software is inadequately maintained due to insufficient or problematic funding for software development and maintenance, as is common with open source projects; and

3) the risk that the software (and the blockchain and other structures built on it) forks, resulting in fractured blockchain networks, due to the customary practice of forking open source software to make desired changes to it.

There are no doubt other critically important operational risks to public blockchains, as I and others have explored (Kiran and Stannett 2014; Peters et al. 2014; Walch 2015). However, in this Chapter, I am focused on the operational risks most related to the use of customary grassroots open source software practices in the running of public blockchains. This analysis marks an expansion of my examination of the operational risks generated by Bitcoin's status as open source software in an earlier paper (Walch 2015), as the topic merits deeper engagement.

Before jumping into the analysis, a very brief primer on open source software is in order.

Open source software is software for which the source code (i.e., the part of the code that is readable by humans) is made freely available to all. It is distinguished from proprietary software, for which the owner of the code does not make the source code available, and for which the owner places limits on use. Open source software comes with a set of core freedoms: "the rights to access the source code, modify the program, and redistribute it, either in its original or modified form." (Nyman 2015, p. 14).

Various practices from the open source software area will be explained as I analyze them in the subsections below. However, one important distinction underlies the entire analysis: whether an open source project is (a) initiated and run by an independent set of software developers (sometimes called an "autonomous" open source project (West and O'Mahoney 2008), though I prefer "grassroots"); or (b) created and run by a legal entity (like a corporation) (sometimes called a "sponsored" or "corporate" open source project). (Nyman 2015, p. 24). Grassroots and corporate open source projects vary significantly in how decisions are made about the code. With a grassroots project, decisions are made through an informal process to reach "rough consensus" (described further below), while in a corporate project, "ultimately the sponsor company…decides what is included in the end product." (Nyman 2015, p. 24). Though there is a spectrum on which different open source project lie, fully grassroots open source lies at one end of the spectrum (where control and power are eschewed), while corporate open source lies at the other end (where control and power are explicit).

Control over decision-making relates to all of the practices I discuss in this Part III, including how the software code changes, how software development is funded, and whether and how the software code is forked. As my analysis will reveal, a lack of defined or accountable control over these processes generates operational risks for public blockchains, impacting their suitability to serve as financial market infrastructure (or other critical societal systems).

A. *Hampered Decision-Making and Grassroots Open Source Software Development Practices*

As noted in Part II, clear governance structures are viewed as essential for FMIs. This makes perfect sense for something that is seen as critical to global financial stability. If something goes wrong with an FMI, a clear chain of command is desirable in order to make decisions quickly, and with accountability. This is a basic tenet of the human experience – that governance structures emerge, and that with high-stakes matters, clarity on responsibility is crucial. This is why there is a long, clear line of succession for the office of President of the United States, and why we insist upon a clear chain of command and well-defined protocols in the military, police and fire departments, hospitals, nuclear reactors and power plants. With high-stakes matters, humans have decided that clear hierarchy and structure are helpful in safe and effective management.

Public blockchains use a very different governance model: the software development process commonly used to develop and maintain grassroots open source software. In this subsection, I describe how this software development (i.e., governance) model hampers the decision-making around changes to a public blockchain's software code.

First, a brief overview of the software development process of grassroots open source software and public blockchains is appropriate.

Decentralized Software Governance

The hallmark of blockchain technology is that it is decentralized – i.e., that there is no central party that maintains this data structure. Public blockchains are decentralized in two ways. First, the network of transaction processors that maintains the ledger is decentralized, and anyone

in the world may freely join this network of computers without needing permission. Second, and more important for our purposes, the governance of the software code that comprises public blockchains is also decentralized and informal. Governance of the software code is extremely important because the code itself is ever-evolving, as new releases of software are issued to fix problems, make improvements, and add new features. With public blockchains, these code changes come about through the efforts of a team of software developers loosely organized under a model typically used for grassroots open source software.

Public blockchains are generally built with open source software. This means that anyone can see, make use of, and make changes to the software code, so long as they make the code they build from it open source. The governance process of open source software is famously informal, with the coders who actually make decisions about changes to the code (known as core developers) gradually rising to the top of the leadership pyramid based on their reputation and performance. The coders of grassroots open source projects do not work for a single organization, and the group of coders working on an open source project may be quite fluid. And, decisions about the code are made based on "rough consensus" rather than a formalized voting or other decision-making process. Further, with grassroots open source projects, coders generally work on the code without compensation, largely because there is no business or entity there to pay them. Contributing to the code is viewed as an altruistic, community-building action within the coding community, so coders usually participate in open source projects as a hobby rather than a full-time job. (I address the risks raised by this funding model in Part III.B below.)

This has been the general software development model used with public blockchains, particularly with Bitcoin. It is worth thinking through the implications of this governance model, particularly in the context of a public blockchain supporting financial market infrastructure (or any other critical public system, really). In considering the governance implications I describe in the following paragraphs, I ask the reader to imagine using this type of governance model with our military defenses (e.g., nuclear weapons) or in an intensive care hospital unit, to concretize how ill-suited this model is for high-stakes matters.

There are a number of ways that the grassroots open source governance model could hamper decisions about the code. First, in this model, no one has the official responsibility for keeping the software operational (Walch 2015). By this, I mean that no one is necessarily accountable for a failure to do so. People choose (or not) to participate in the software development process, and have complete freedom (without consequence, other than possible reputational harm) to help or not help in a moment of crisis. Developers who have previously maintained the software are under no obligation to act in a crisis, and may find it riskier to act than to abandon the system. While core developers have acted in the past to resolve crises with the Bitcoin and Ethereum blockchains (the March 2013 fork, for Bitcoin, and The DAO theft, for Ethereum), there is no guarantee that they would do so in the future (Walch 2015).

Second, under the grassroots open source governance model, no one is in charge of making decisions for the network. I have argued previously:

> "As there is no defined power or accountability structure, no one has to listen to anyone else's ideas about how to resolve a crisis. There are no definitively appointed decision-makers. This is different from having no one responsible for keeping the software operational; this risk is that even if people decide to take on responsibility for resolving a

> problem with the …software…, their authority to do so, and their resulting ability to implement their solution, is in question. This means that anyone with a suggested resolution to a crisis may merely propose a solution, but it may take too long to achieve buy-in from other members of the … community to successfully implement the solution in an emergency situation. We see this type of argument commonly made in debates over the limits of the executive power of the President of the United States, who may need to act quickly in a crisis without waiting for specific authority from Congress" (Walch 2015, p. 871).

Third, this amorphous governance model can lead to unacknowledged centralization of power, resulting in unaccountable or unchecked power. The core developers of public blockchains are more powerful than the rank-and-file developers on these projects. In Bitcoin, for instance, a small number of core developers are the only people who have the passwords to actually enter changes into the underlying code (known as having "commit access"). They also act as the voice of the blockchain through their interactions with the media, regulators, and others in the blockchain ecosystem, as their recommendations and insights are seen as relevant and consequential to the future of the applicable blockchain. As an example, many core developers are frequent panelists or keynote speakers at blockchain or fintech conferences around the world.

What is problematic about unacknowledged centralization of power is not the *centralization* part, but the *unacknowledged* part. Unacknowledged, or hidden, power, can lead to the exercise of unaccountable, unchecked power. With precisely delegated power, it is clear what actions one can and cannot take, but with amorphous powers, the limits of power are fuzzy, and can easily be expanded. Unaccountable power is a bad fit for financial market infrastructures, or for other critical public systems. (The unaccountable power that can arise in these structures is, ironically exactly what these open systems were designed to fight against, as they are reactions to the closed (unaccountable) software development process for proprietary software.)

All of these scenarios described in this Part III.A could either paralyze or delay critical decisions about the software code, endangering all structures built on top of it. Indeed, a debate is ongoing in the public blockchain world over which software changes should be considered purely technical versus those considered more ideological, and these debates create the potential for software forks, as I describe in Part III.C below. Moreover, if someone does act as if he or she has authority (similar to Vitalik Buterin of Ethereum), there is a chance that the decision will not receive buy-in from the blockchain community, again potentially leading to forks in the code and blockchain.

As with law, change is necessary to all software in order for it to continue to be useful. If a software governance process generates paralysis, the software cannot improve or adjust to changing conditions. The balance between concentrated and distributed power is difficult to strike, but the standard grassroots open source software development process appears too far along the spectrum of (nominally) distributed power to govern critical systems like financial market infrastructure.

Perhaps in recognition of this problem, newer public blockchains appear to be tweaking the typical informal open source governance process, adding more structure. Z-cash, a

cryptocurrency launched to wide interest in October 2016, is based on the Bitcoin code, but with enhanced privacy through 'zero-knowledge proofs.' It has established a slightly more formal governance structure than that used on Bitcoin, but analysis of that structure and its implications will have to wait for a future paper. Ethereum, another public blockchain, has also adjusted governance, relying heavily on founder Vitalik Buterin to guide the trajectory of the project. Z-cash also has a founder, Zooko Wilcox, who is strongly identified with the project. Crucial to note here is that the governance structures of Z-cash, Ethereum, and others are experiments, or works-in-progress, and it is unclear whether they will function better than the purer grassroots open source software development process used in Bitcoin. It is one thing to experiment with a new type of technology for financial market infrastructures, but another level of risk is added when the governance of the technology is also experimental.

> Given that the global standards for financial market infrastructures state that
>
> "FMI[s] should have governance arrangements that are clear and transparent, promote the safety and efficiency of the FMI, and support the stability of the broader financial system, other relevant public interest considerations, and the objectives of relevant stakeholders" (*PFMI* 2012, p. 1),

it is unlikely that using informal, experimental, grassroots open source governance practices in public blockchains could satisfy this standard.

B. *Inadequate Software Maintenance and Problematic Open Source Funding Model*

The second common open-source software practice I examine is how open source software development is funded. It is widely acknowledged that funding grassroots open source software development is very difficult, as it relies on coders to contribute to the code without pay, or to find alternative sources of funding that may raise conflict of interest questions. In this section, I explore how relying on the traditional open source software development model to fund public blockchains exposes them to the operational risk of inadequate attention to software maintenance and development, or to particular interests shaping the trajectory of these public structures. This risk is problematic for any public blockchain that serves as the backbone of financial market infrastructure (or any other critically important public system).

As mentioned earlier, one of the celebrated attributes of open source software is that those who develop the software code generally do so without compensation. Developing free open source software is seen as an altruistic or reputation-enhancing activity among the coding community, and is often done by software developers outside of their regular paid employment. This is part of the ideology of the open source software movement, and it has been successful with many types of software.

There has been a slowly dawning realization, however, that this funding model may be a bad fit for critically important software. Following the 2014 discovery of the catastrophic Heartbleed bug in Open SSL (an open source software that runs a key security layer of the Internet), a group of technology companies formed the Core Infrastructure Initiative to better support the development of critical open source software projects. Many open source software projects have only a few active developers, when a much more substantial dedicated team of coders is needed to adequately maintain the software (Wheeler and Khakimov 2015). Inadequate attention to the

code over time increases the likelihood that bugs aren't seen and fixes aren't made. The Core Infrastructure Initiative is raising funds from its members to pay developers on various open source projects that are deemed to have a critical need. Mozilla, a prominent company that maintains certain open source software like the Firefox web browser, recently formed the Secure Open Source ('SOS') project to provide funds to increase the security of selected open source projects. This initiative grew out of a 2015 Mozilla research project that involved surveying cybersecurity experts about key threats to cybersecurity. The report for the project noted that

> "Participants saw the funding of security audits of critical open source projects as a key unresolved and priority issue in cybersecurity policy. Indeed, funding for free and critical open source projects emerged as an interesting outlier in becoming the one issue perceived by all as both highly desirable and feasible in a government cybersecurity policy agenda" (Francois et al. 2016, p. 15).

Bugs continue to be found in critical open source projects, including the critical vulnerability termed "Dirty Cow" discovered in the Linux kernel in October 2016.

The open source software funding dilemma plagues the software development process for public blockchains as well. When Bitcoin was created as free open source software by the mysterious "Satoshi Nakamoto" back in 2008/2009, it was of little if any significance to the public. Beginning with a community of one (the creator), it gradually spread through a group of early adopter coders, spending years wandering in the wilderness before it gained widespread attention around 2013. And the cryptocurrency exchanged on the Bitcoin blockchain had little value for a very long time, only gradually moving from a few cents per bitcoin to a few dollars, to its explosion in value in 2013. Thus, for the first several years of its life, Bitcoin was a low-stakes project, a game for the early participants in the system. It was fine for the early coders to work on the software as a hobby because there was little money at stake for them or anyone else. No one would lose much if the system failed altogether. It was just a really interesting experiment.

The stakes changed dramatically once more of the public became aware of Bitcoin and began to see its usefulness. And, as speculators entered the market and the mining sector professionalized from one guy with a computer in his bedroom to vast server farms strategically placed around the world, the stakes continued to increase. Suddenly, it mattered a great deal if there was a bug in the code, or if the software had not been optimized to run most efficiently. The software had to run smoothly 24/7, and coders had to respond to crises on an emergency basis in this now mission-critical system. Unsurprisingly, it became impossible for key developers to run an always-on mission-critical system as an unpaid hobby. Seeing the need for dedicated attention to the code, companies within the Bitcoin ecosystem (e.g., BitPay, Blockstream) began paying some core developers. Several non-profits (The Bitcoin Foundation, MIT) also stepped up to fund the developers.

Public blockchains that have been introduced since Bitcoin gained mainstream recognition do not have the same chance to make unnoticed mistakes in their youth. They are potentially high-stakes from the day they are launched, as they purport to facilitate the exchange of value for members of the public. This means that expecting developers to run these systems for free in

their spare time is a significant risk. Recognition of this problem has spawned creative ways to fund the software developers for Bitcoin and other public blockchains. With Ethereum, the software developers have been compensated by a "pre-sale" of ether, the currency of the Ethereum blockchain, and finances appear to be managed by the Ethereum Foundation, a Swiss non-profit set up specifically to offer financial and advisory support to the development team. With the recently released Zcash blockchain, the developers will fund themselves through the issuance of "tokens" that will trade on the blockchain (in effect, acting as an issuer of money that will be used in this particular blockchain community). A flurry of so-called "app coins" have sprung up in 2016, funded, like Zcash and Ethereum, through the sale of tokens by developers. Called an "Initial Coin Offering" or ICO, this funding model has drawn scrutiny from securities attorneys, who warn that the issuance of these tokens without registration may violate the securities laws (Byrne 2016).

The different approaches to the problem of funding open source software development are creative, but more research into the implications of the funding methods, as well as their stability over the long term, is needed. As I have discussed elsewhere, these private funding structures create potential conflicts of interest in these public structures (Walch 2015). If developers are tied to a particular funding source, there is the chance that the developers will be influenced by the people who are paying them, rather than by the interests of the people using or relying on the blockchain. Further, there is the question of whether developer funds will be around in the long-term, or whether they could be cut off if the funder loses interest or it becomes politically controversial to fund a particular blockchain. This is problematic in a public structure like a public blockchain, particularly if it comes to underlie financial market infrastructures or other critical systems.

* * *

As discussed in this sub-part, the use of grassroots open source software funding methods creates operational risks for public blockchains, which negatively impact their suitability to support financial market infrastructures.

C. Fractured Networks Caused by Open Source Software Forking Practices

The final open source practice that I consider in the context of public blockchains is that of forking software code. Anyone who has followed the blockchain world recently has been made vividly aware of the possibility that a public blockchain could fork into two (or more) separate networks (and accompanying ledgers), as the Ethereum blockchain did in dramatic fashion during the summer of 2016 (and as I discuss later in this Part III.C).

In this subpart, I will explain the practice of forking in open source software, provide examples of how this phenomenon has manifested in public blockchains, and explicate the operational risks this practice raises for public blockchains.

First, what is "forking" in open source software? As Nyman noted in his recent work on the topic, "a very general interpretation of what forking means is copying an existing [software] program and distributing a modified version of it." (Nyman 2015, p. 1). If source code is publicly available, and can legally be changed by anyone, then open source software code is inherently forkable. This is in sharp contrast to proprietary software, which cannot be forked by anyone

other than its owner, both because the source code is not made publicly available, and because legal restrictions forbid it. But, "with open source software, one cannot *forbid* anyone from forking the code." (Nyman 2015, p. 1, original emphasis).

Fascinatingly, there is little academic research on code forking, with Linus Nyman's 2015 dissertation, *Understanding Code Forking in Open Source Software*, the first wide-ranging academic work to focus on this important practice. Nyman's work reveals that there are pros and cons to the forking phenomenon, with sustainability of the software one potential plus and complexity and confusion a potential negative, "with forks spawning forks of their own that, in turn, may be forked, and forked again." (Nyman 2015, p. 6). Forking of significant open source software projects, such as GNU/Linux or MySQL, has been relatively rare, but the potential outcomes that Nyman notes are significant in evaluating the operational risk profile of public blockchains. These outcomes are: (a) peaceful co-existence of both old and new software; (b) the old version of the software dies; (c) the new version of the software dies; or (d) there is a contentious co-existence of the old and new software. With public blockchains, it is not just software that forks, but entire networks, making the forking option even more consequential.

In Bitcoin, for instance, there are different types of forks that can occur through new releases of software, with varying effects on the network (bitcoin.org 2016). Hard forks are the most extreme, in that they can create competing versions of the blockchain when nodes within the network run different versions of the software. This makes it highly desirable that all (or a great majority of) nodes upgrade to each new release.

Although forks are part and parcel of open source software, with public blockchains, forking appears to have much graver consequences than it does in other forms of open source software. This is due to several unique attributes of public blockchains: the fact that they purport to actually embed and transfer value, and the fact that they purport to serve as an authoritative record of events (whatever those events may be). If these structures fragment, there is no longer a *single* authoritative data structure, but *many*, greatly undermining the technology's service as a single, reliable source of truth.

Below, I provide three real-world examples from the public blockchain world that demonstrate some potential consequences of the forking possibility: 1) the March 2013 hard fork in the Bitcoin blockchain; 2) the "block size debate" within the Bitcoin community, ongoing since summer 2015 and still unresolved; and 3) the hard fork in the Ethereum blockchain in July 2016.

March 2013 Hard Fork

In March 2013, a hard fork unexpectedly occurred in the Bitcoin network. This meant that two different versions of a distributed ledger were being recognized as accurate by different portions of the network. In essence, the network had fractured in two, meaning that there were also two distinct ledgers being maintained.

The cause of the fork was the use of different versions of software by the computers that operate the network. Some computers had upgraded to a new version of the software, while others had not This is not an unlikely occurrence in a system of disaggregated computers, whose owners cannot be compelled to adopt new versions of the code.

What to do when the Bitcoin network is split in two? Which tokens on the ledgers are bitcoins and which are something new? The existential chaos spawned by the fork was quickly recognized, and the community of software developers and transaction processors went to work to pull everyone back onto the same chain. This required getting a certain portion of the computing power to agree to use the previous version of the software so enough of the network was running the same code. Clearly, the most efficient way to achieve this would be for a few holders of big chunks of computing power to downgrade, rather than asking individuals who held a miniscule portion of the network's power. So, the core developers went after the big-enough fish, asking them to forego amounts they had been paid for performing transaction processing services on one chain, and switch to the other chain. The switchers had to *sacrifice their own earnings for the benefit of the Bitcoin network as a whole* – i.e., act altruistically for the greater good.

Through these frantic efforts, the severed networks were reunited. But, how was the surviving chain selected? Ironically, it was by the core developers, in this system that purports to have no humans in charge and to operate purely through the power of code and mathematics.

Bitcoin Block-Size Debate

The Bitcoin community learned a lot from the March 2013 fork, and has been extremely skittish of a hard fork ever since. Hence, we have seen the long-running drama known in the industry as "The Block Size Debate."

The Block Size Debate is a dispute over how large a 'block' within the Bitcoin blockchain should be. This is part of a larger discussion of how the Bitcoin software and network must change to accommodate a higher number of transactions per second, as it must if it were to become more widely used. Different factions of software developers have introduced various proposals for how to scale up the network, but the issue has remained unresolved since 2015 – over a year as of this writing. The general consensus is that a hard fork would be necessary to implement certain proposals to scale the network, making the decision fraught with risk.

Though the size of a block would seem to be a purely technical question, with the answer determined by weighing the technical characteristics of the network, the debate has revealed that it is very much a political question, with implications for the purposes and values of the Bitcoin technology generally. This is because the resolution of the question may affect how expensive it is to participate in processing the transactions on the network, meaning the network could become more and more centralized as costs to participate increased. As the Bitcoin network was created as an expression of a particular political philosophy (libertarianism leaning toward anarchism), there is a contingent of developers who feel strongly that the network needs to remain as decentralized as possible to be true to its founding principles. Others feel that the principles of the network need to move with the times (echoing the debates over whether the U.S. Constitution should be hewn to as it was intended by its drafters, or should live and flex over time with societal changes).

The debate has been impassioned, with shifts in the cast of characters, prominent developers publicly renouncing Bitcoin, and international summits to try to reach agreement, but the network has not been able to move forward with a resolution. Diplomacy and public relations skills have become vital as developers try to persuade the large mining pools (many in China)

that their solution to the problem is the best. This is because the software change cannot be implemented without at least 51% of the computing power (provided by the miners) adopting it.

And, probably in large part because the consequences are so extreme, the debate has paralyzed the Bitcoin community. Neither side can be guaranteed to win enough votes, so the election remains unheld. Estimates of the percentage of the computing power that have to adopt a new release for a Bitcoin hard fork to be deemed a success vary, but the July 2016 Ethereum hard fork (discussed below) has revealed that even a small number of holdouts can result in a competing blockchain.

In some ways, the situation is similar to the inertia that grips major social programs such as Medicare or Social Security. There is general agreement that significant change is needed to these programs, but the change is difficult to push through, in part because the transition will be so difficult. Here, the consequence is that any real change to the software can completely shatter the system into split chains, making it as fragile as a brittle set of bones.

July 2016 Ethereum Hard Fork

A third example of the forking risk played out during the summer of 2016 with a controversial hard fork of the Ethereum public blockchain.

The saga began with the creation of The DAO, a "decentralized autonomous organization" built on top of the Ethereum blockchain. Designed as an automated venture capital fund for blockchain investments, The DAO drew around $150 million in investments, but was hacked shortly after its launch, resulting in the transfer of $60 million of ether (the currency of the Ethereum blockchain) to its attacker. As The DAO's premise was that it was "unstoppable code" with which no humans could interfere, the Ethereum core developers were faced with two undesirable options: (a) do nothing about the hack because it was merely an exploit of the software code every investor in The DAO had agreed to, creating a black eye for the technology, and potentially opening Ethereum and DAO coders up to lawsuits; or (b) issue a new release of Ethereum software that would remedy the theft by taking back the hacked ether, undermining the Ethereum blockchain's claims to be immutable and demonstrating the centralized power possessed by the core developers.

Ultimately, the core developers decided to recommend the hard fork, which required them to persuade the miners of the Ethereum network to adopt the newly-released software. They persuaded most, but not all, to upgrade to the new software. The end result was that Ethereum split into two separate blockchain networks: (a) the Ethereum network that adopted the new software release, and (b) the Ethereum network that did not (known now as Ethereum Classic). Each now has its own set of core developers and miners, and seems to operate independently from the other. These blockchains are identical up to a certain point, and then diverge in content, which means that any system built atop the Ethereum blockchain prior to the fork had to make a decision about which blockchain to remain on.

And, as one might expect after the Ethereum developers' intervention, the question of whether a public blockchain is or should be immutable has become a hot topic in blockchain circles, and since July 2016, Ethereum has had to hard fork the software several times to repair serious software bugs (Hertig 2016).

51% Attack Risk

Finally, the forking risk is complicated further in public blockchains through the 51% Attack risk. Public blockchains, as currently structured, are vulnerable to the risk that participants in the transaction processing (mining) network could monopolize decisions about the path of the network, including potentially adopting new forms of software, revising previous entries on the blockchain, or preventing new entries from selected (or any) parties from being entered on the blockchain. This is because whatever 51% of the transaction processing network decides to do, is done, as the networks run through majority rule.

The vulnerability of a network to a 51% attack can increase immediately following a fork, as the computing power previously devoted to the single 'parent' network becomes split between the two 'child' networks. This means that less computing power is needed to attack each of the surviving networks. This played out in the immediate aftermath of the July 2016 Ethereum hark fork, with a threat by a miner to attack the Ethereum Classic network (Quentson 2016).

Lessons Learned

The forking-related events described above reveal a number of important truths about public blockchains, and in this subsection, I reflect on what these episodes can teach us.

First, the 2013 fork in the Bitcoin blockchain demonstrates that new software releases for public blockchains can lead to fractured networks. A fork into separate blockchains can happen when a new release is incompatible with earlier releases, and, because there are no forced software updates in a public network, there is no way to guarantee that all members of the network will move to the new version of the software in a timely manner. In part, fear of a forked network is driving the paralysis of Bitcoin in the never-ending block size debate.

Second, the 2013 Bitcoin fork also reveals that rejoining forked blockchains may require human coordination (amongst the developers and the miners), as well as a willingness on the part of certain miners to sacrifice their earnings on one blockchain as part of rejoining the other blockchain.

Third, both the 2013 Bitcoin fork and the July 2016 Ethereum Hard Fork show that the core developers of public blockchains wield significant power in identifying and remedying a fork, in that they can coordinate communications within the mining network, and influence which chain survives (although the existence of Ethereum Classic shows they cannot necessarily eliminate a competing chain).

Fourth, the Bitcoin block size debate demonstrates how the risk of a forked network can paralyze a public blockchain, potentially leaving significant problems with the code unsolved because the appropriate solutions are disputed. Certain miners in the Bitcoin network have emerged as holdouts on various proposals, indicating how difficult it is to force consensus on a controversial software change.

Fifth, the July 2016 Ethereum Hard Fork shows that events related to processes built on top of public blockchains can influence decisions about changes made to the blockchain software itself. The DAO's problems were not a problem with the Ethereum blockchain, but with The DAO's code, yet led to a hard fork in the Ethereum blockchain through a new Ethereum software

release that essentially erased The DAO theft. This means that disparate applications and processes built upon a public blockchain may be impacted by decisions made by other applications built on that blockchain, in addition to decisions made about the underlying blockchain itself. In The DAO episode, other applications built on the Ethereum blockchain were affected by Ethereum's hard fork, even though the fork was driven by events associated solely with The DAO.

Sixth, The DAO hack and the resulting Ethereum hard fork are reminders that our human imperfections manifest in the software code that we write. Many have suggested that software code can automate governance, yet humans cannot write flawless code, meaning that human interventions will remain necessary even with automated technologies. Because we cannot predict the future perfectly, there will always be risks we cannot anticipate, so our governance systems must maintain at least a hint of flexibility, much as legal contracts and laws themselves are amendable to fix errors or adjust to new circumstances.

Seventh, the Ethereum hard fork and the ongoing existence of Ethereum Classic demonstrate that competing blockchains can result from a hard fork. This phenomenon raises many questions, including practical ones such as which of the splintered chains to recognize and treat as legitimate, and how legal rights and liabilities tied to processes built on top of the parent chain play out when the parent chain splits in two. For instance, which series of trading records is legitimate, when there are suddenly two networks? Given that forks can spawn forks can spawn forks (*ad infinitum*), this could become rather complicated to manage, with each fork raising potentially contentious legal and economic issues. The forking of a blockchain network is analogous to a spin-off of a company, which is an enormously complicated process requiring careful attention to details to ensure that rights and obligations are appropriately defined and separated. Thus, any systems built atop public blockchains, including financial market infrastructures, may have these complexities sprung on them at any time through hard forks, greatly reducing any control these systems have over their risk exposures.

Eighth, all of the learnings I have described here cumulatively point to how public blockchains magnify normal software risk for processes built on them, and would do the same for any financial market infrastructures that relied on them. This is because public blockchains attempt to yoke their participants together in ways that other software does not – the *chain* of blockchain technology binds its users as well as the data stored in the ledger.

At a fundamental level, blockchain technology's benefit is that it keeps everyone in the network on the same (metaphorical) page. Each person has a real-time, correct version of the shared ledger. For the system to continue to have value, everyone must remain on the same page. Running the software and participating in the network means that one is committing to staying on the same page as other network participants. (Thomas (2016) similarly explores this concept of the problems raised by maintaining a shared state in blockchains.) A 'single-member blockchain' is an oxymoron, as a blockchain is inherently a group activity, intended to memorialize the relevant actions of its participants.

With the possibility of software forks inherent to open source software, a public blockchain's network cohesion (and entire value proposition) is threatened every time a non-backwards-compatible software release is proposed and unevenly adopted by the network. Each proposal for a hard fork is analogous to calling for a binding referendum on secession from the

blockchain, with votes cast through the choice of upgrading to the new software release (or not). Continuing with the real-world referendum analogy, if one is in the minority of computing power that chooses not to adopt (vote for) the new release, one has essentially seceded from the blockchain. Suddenly, those on the left-behind chain have to find their own resources (developers, miners, etc) to continue to function, just as Ethereum Classic has had to. So, each proposed hard fork is incredibly high-stakes, as evidenced by the agonizing Bitcoin block size debate.

Although the open source software model has been highly successful in many instances, the forking possibility may make it unsuitable for public blockchains, at least if these blockchains undergird financial market infrastructures or other important societal systems. When we purport to embed actual value or records of group events in blockchain technology, it becomes qualitatively different from other software. Although my theory about this phenomenon is still taking shape, as I have explained here, I believe it has to do with tying the participants in the network together for every step that is taken. Those who break free of the chain must be willing to build a new system for themselves, exposing systems built atop the parent chain to these shifting foundations.

IV. REFLECTIONS

This chapter seeks to contribute to the discussion of the risks that certain practices common to open source software raise for public blockchains. In this chapter, I have focused on their potential role as the technology undergirding financial market infrastructures, whose uninterrupted operation is critical for global financial stability.

The reliability and certainty that the financial sector sees in blockchain technology is undermined in public blockchains by the use of traditional grassroots open source software development processes, including informal governance, problematic funding, and the potential for software forks. These practices create operational risks for public blockchains, making them less solid than they are often said to be.

Of course, each public blockchain has its own particular characteristics, and thus a different overall risk profile. All share the exposure to forks, as that is an inherent characteristic of open source software. Some may have more or less structured software development methods, and some have formalized the funding of software development through a non-profit foundation or through the issuance of a percentage of the applicable cryptocurrency to the founding development team. Each of these choices affects where the blockchain falls on the risk spectrum.

However, even with various tweaks to each of the practices outlined in this paper, public blockchains operate very differently from how we expect critical infrastructures to. As demonstrated with the regulations for financial market infrastructures, clear governance, comprehensive risk management, and identifying and mitigating operational risks are essential to managing these important structures. And, as outlined in Part III of this chapter, the grassroots open source software practices associated with public blockchains are diametrically opposed to the more controlled practices we expect in high-stakes areas.

In a broader sense, the analysis in this chapter suggests that it may be time for a rethink about the role of grassroots open source software in critical infrastructures outside the blockchain technology setting. Open source software performs many infrastructural functions in our society, including processes crucial to the operation of the Internet. As with Bitcoin, practices that worked fine when the project was small-scale and low-stakes may be inappropriate for large-scale, high-stakes projects. We are discovering now that many critical open source software projects, undergirding vital pieces of the Internet, are understaffed, underfunded, and insecure. We may be in the process of discovering why these revolutionary processes (loosely structured governance, unpredictable funding, forking as an option) have not been widely adopted for critical public practices. It may be that we just can't be comfortable enough with them to count on them in a crisis. Analogously, volunteer fire departments are relatively common in small towns, but cities pay fire departments to fulfill this important public function – the scale of the systems seems to dictate a more formal structure being needed in the more populous cities.

To open source software devotees, these observations may be viewed as fighting words. Open source is as much an ideology as it is a technical practice, and any critique of it inspires passionate defense by its adherents. The superiority of open source software to proprietary software—pretty much regardless of the task or setting—is treated as dogma by open source advocates, and, as critics of religion have long seen, questioning dogma can be dangerous.

Yet we cannot fully evaluate our practices unless we are able to question our most basic assumptions. The courage to question existing practices gave rise to Bitcoin itself, and continued questioning and critique will help us to responsibly use the underlying blockchain technology. Indeed, our infrastructures depend on it.

REFERENCES

Arnold, M. (2016). Accenture to unveil blockchain editing technique. *Financial Times*. 19 September 2016.

Bank for International Settlements Committee on Payments and Market Infrastructures and Board of the International Organization of Securities Commissions (2015). *Consultative Report: Guidance on cyber resilience for financial market infrastructures*. November 2015. Available from: http://www.bis.org/cpmi/publ/d138.pdf [Accessed 29 November 2016].

Bank for International Settlements Committee on Payment and Settlement Systems and Technical Committee of the International Organization of Securities Commissions (2012). *Principles for Financial Market Infrastructures.* April 2012. Available from: http://www.bis.org/cpmi/publ/d101a.pdf [Accessed 30 November 2016].

Bitcoin.org (2016) *Bitcoin Developer Guide.* Available from: https://bitcoin.org/en/developer-guide [Accessed 4 December 2016].

Burton, B. and D. Willis (2016). *Gartner's 2016 Hype Cycles Highlight Digital Business Ecosystems* Available from: www.gartner.com [Accessed 25 November 2016].

Byrne, P.J. (2016). Against Tokens (and Token Crowdsales). *The Back of the Envelope (a blog)*. 12 August 2016. Available from: https://prestonbyrne.com/2016/08/12/against-crowdsales/ [Accessed 1 December 2016].

Carlsten, M. et al. (2016). On the Instability of Bitcoin Without the Block Reward. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. Pp. 154-167. 24 October 2016

Carney, M. (2016). *Enabling the FinTech transformation: Revolution, Restoration, or Reformation?* (Speech that was to have been given by Mark Carney, Governor of the Bank of England at the Lord Mayor's Banquet for Bankers and Merchants of the City of London at the Mansion House, London). 16 June 2016. Available from: http://www.bankofengland.co.uk/publications/Documents/speeches/2016/speech914.pdf [Accessed 29 November 2016].

DTCC (2016). *Embracing Disruption: Tapping the Potential of Distributed Ledgers to Improve the Post-Trade Landscape*. January 2016. Available from: http://www.dtcc.com/news/2016/january/25/blockchain-white-paper [Accessed 29 November 2016].

Eha, B.P. (2016). MUFG Aims to Use Bitcoin to Improve Cross-Border Payments. *American Banker*. 28 October 2016.

Federal Reserve (2016) *Policy on Payment System Risk*. 23 September 2016. Available from: https://www.federalreserve.gov/paymentsystems/files/psr_policy.pdf [Accessed 30 November 2016].

Francois, C. et al. (2015). The Mozilla Cybersecurity Delphi 1.0: Towards a User-Centric Policy Framework. July 2015. Available from: https://blog.mozilla.org/netpolicy/files/2015/07/Mozilla-Cybersecurity-Delphi-1.0.pdf [Accessed 1 December 2015].

Giancarlo, J.C. (2016). *Regulators and the Blockchain: First, Do No Harm* (Special Address of CFTC Commissioner J. Christopher Giancarlo Before the Depository Trust & Clearing Corporation 2016 Blockchain Symposium). 29 March 2016. Available from: http://www.cftc.gov/PressRoom/SpeechesTestimony/opagiancarlo-13 [Accessed 29 November 2016].

Hertig, A. (2016). How Developers Are Responding to Ethereum's Unexpected Fork. *CoinDesk.* 1 December 2016. Available from: http://www.coindesk.com/developer-response-ethereum-fork/ [Accessed 2 December 2016].

Kaminska, I. (2016) Blockchain and the holy real-time settlement grail. *Financial Times* [online]. 26 February 2016. Available from: https://ftalphaville.ft.com/2016/02/26/2154510/blockchain-and-the-holy-real-time-settlement-grail/ [Accessed 1 December 2016].

Kiran, M. and M. Stannett (2014). Bitcoin Risk Analysis. *NEMODE*. Available from: http://www.nemode.ac.uk/wp-content/uploads/2015/02/2015-Bit-Coin-risk-analysis.pdf [Accessed 1 December 2016].

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.* Ph.D. Thesis, Hanken School of Economics.

Perlroth, N. and M. Corkery (2016). Details Emerge on Global Bank Heists by Hackers. *The New York Times*. 13 May 2016. Available from: http://www.nytimes.com/2016/05/14/business/dealbook/details-emerge-on-global-bank-heists-by-hackers.html?_r=0 [Accessed 29 November 2016].

Peters, G.W. et al. (2014). Opening discussion on banking sector risk exposures and vulnerabilities from virtual currencies: An operational risk perspective. *arXiv.org*. Available from: https://arxiv.org/ftp/arxiv/papers/1409/1409.1451.pdf [Accessed 1 December 2016].

PWC (2016). Bank of England FinTech Accelerator partners with PWC on distributed ledger Proof of Concept. 17 June 2016. Available from: http://pwc.blogs.com/press_room/2016/06/bank-of-england-fintech-accelerator-partners-with-pwc-on-distributed-ledger-proof-of-concept-.html [Accessed 29 November 2016].

Quentson, A. (2016). Miners to Attack Ethereum Classic after Poloniex's Listing. *Cryptocoins News*. 24 July 2016. Available from https://www.cryptocoinsnews.com/miners-attack-ethereum-classic-polonixs-listing/ [Accessed 1 December 2016].

Rapier, G. (2016). Yellen Reportedly Urges Central Banks to Study Blockchain, Bitcoin. *American Banker*. 7 June 2016.

Rizzo, P. (2016). Linux, IBM Share Bold Vision for Hyperledger Project, a Blockchain Fabric for Business. *CoinDesk*. 11 February 2016. Available from: http://www.coindesk.com/linux-ibm-hyperledger-blockchain-business/ [Accessed 29 November 2016].

Thomas, S. (2016). The Subtle Tyranny of Blockchain. 18 August 2016. Available from: https://medium.com/@justmoon/the-subtle-tyranny-of-blockchain-91d98b8a3a65#.l4jt4z2ze [Accessed 1 December 2016].

Walch, A. (2015). The Bitcoin Blockchain as Financial Market Infrastructure: A Consideration of Operational Risk. *NYU Journal of Legislation & Public Policy*, vol. 18, no. 4, pp. 837-893.

West, J. and S. O'Mahony (2008). The Role of Participation Architecture in Growing Open Sponsored Open Source Communities. *Industry and Innovation.* Vol. 15, no. 2, pp. 145-168.

Wheeler, D.A. and S. Khakimov (2015). Open Source Software Projects Needing Security Investments. (White Paper of the Institute for Defense Analysis and the Linux Foundation). 19 June 2015. Available from https://www.coreinfrastructure.org/sites/cii/files/pages/files/pub_ida_lf_cii_070915.pdf [Accessed 1 December 2016].

White, M.J. (2016). *Opening Remarks at the Fintech Forum.* (Public Remarks delivered by SEC Chair at SEC Fintech Forum). 14 November 2016. Available from: https://www.sec.gov/news/statement/white-opening-remarks-fintech-forum.html [Accessed 1 December 2016].