Patterns of Self-Organising in the Bitcoin Online Community: Code Forking as Organising in Digital Infrastructure

Completed Research Paper

Jonas Valbjørn Andersen
VU Amsterdam
De Boelelaan 1105, 1085 Amsterdam
j.v.andersen@vu.nl

Claire Ingram Bogusz
Stockholm School of Economics
Sveavägen 65, 113 83 Stockholm
claire.ingram@hhs.se

Abstract

Digital infrastructures play an increasingly central role in shaping existing organisations and creating new ones. Research on digital infrastructure has rested on the assumption that infrastructures are developed to support pre-existing organised activities. However, with the rise of new digital infrastructures supporting open source projects and blockchain communities such as Bitcoin, development of the technological infrastructure also gives rise to a new way of self-organising. Specifically, forking of the underlying source code and subsequent community adoption is increasingly observed to trigger new patterns of self-organising. In order to explore and develop this concept, this paper investigates a case of such distributed digital community: the emergence of the Bitcoin community around a specific instantiation of the Blockchain infrastructure. Our study examines how the community emerges, and how changes in the source code lead to different patterns of self-organising. The paper develops a conceptual framework of self-organising in distributed communities emerging around digital infrastructures.

Keywords: Bitcoin, Blockchain, Online communities, Digital infrastructure, Self-organising

Introduction

The Bitcoin community belongs to a new breed of organisation: without offices, managers, contracts, policies or payrolls, and without strategies, charters or business plans, these organisations are fluid and digital in nature (Barrett et al. 2016). In this particular case, an organisation emerged around a digital infrastructure, known as the Blockchain, and was shaped by the online activities of a self-organising community of distributed individuals, known as an Open Source (OS) community.

Interest in the Blockchain has grown in recent years; where once it was largely known for its role in automating transactions made using the cryptocurrency Bitcoin, it is today being developed for other purposes, including the transfer of other kinds of assets, and for recordkeeping (Morisse and Ingram 2016). The original Bitcoin Blockchain, however, was not built to support these kinds of individual or organisational aims. Although its founder(s), pseudonymous Satoshi Nakamoto, discussed in a white paper how it might revolutionise the finance industry, it was not developed by an organisation with the intention of changing the industry, merely of showing how this might be done (Nakamoto 2008). Moreover, its founder(s) withdrew from the development of the project at a very early stage—leaving a new community to form around it. As the infrastructure pre-dated the community, it drove how the community developed and was organised. Evolution of the infrastructure was ultimately decided by community members' adoption of pieces of code. However, they could not use the infrastructure for anything other than its original sets of functions without changing it considerably, and these changes were constrained by elements of the infrastructure's source code. This constraining function of code has not been as visible in infrastructures that have previously been studied (Iannacci 2010; Kuk and Janssen 2013).

The underlying source code puts limits on what members of the community can do. For instance, the entry of a new transaction onto the blockchain by a miner is communicated to the other miners in the network in order to for them to verify that it is legitimate and consistent with previous entries (and does not come from a fake account, for instance). In this way, the Bitcoin infrastructure is both kept up to date, and its contents are verified and stored by other miners. The software is designed so that transactions can only be added onto the blockchain after verification by the rest of the actors, and cannot be removed once entered without changing the entire blockchain.[1] The blockchain therefore becomes more-or-less unassailable. This position is secured by virtue of a part of the source code in the Blockchain protocol, which says that the version of the software, which includes the blockchain, held by the majority of miners is the "real" Blockchain (Nakamoto 2008; Taylor 2013).

This code implementation prevents individual actors from changing the blockchain. However, it also has another effect: in order for large changes, known as code forks, to be made to the Blockchain, the majority of 'miners' (community members that process transactions) has to adopt them. When this occurs, those miners running the version that is in the minority are seen to be running a *de facto* alternative. That is, they are no longer running a compatible version of the infrastructure—neither the source code that they run nor the transactions entered into minority-held alternative blockchain will be recognised by the original source code. This is, however, only true when the versions are inconsistent with one another; more subtle implications apply when minor updates of the code or consistent code additions are involved.

In their seminal 1996 paper, Star and Ruhleder outline what scholars of digital infrastructures today consider to be a foundational understanding: infrastructure is something that "…becomes infrastructure in relation to *organised practices*" (1996, p.112, emphasis ours). Consequently, one of the explicit characteristics of infrastructure is that it relies on established organisational practices. This notion is echoed in studies of digital infrastructures; Henfridsson and Bygstad, citing Malhotra et al (2004) for instance, describe them as "…partner interface-directed information systems that enable an enterprise to process information collected from its supply chain partners so as to create new knowledge" (Henfridsson and Bygstad 2013, p. 909). Indeed, this sentiment is expressed as one of the explicit assumptions underlying the study of infrastructures: "Infrastructure does not grow *de novo*; it wrestles the inertia of the installed [organising] base" (Star and Ruhleder 1996, p. 113).

However, research on digital infrastructure has also highlighted the generative capacity of digital infrastructures to transform organisations (Hanseth and Aanestad 2003; Henfridsson and Bygstad 2013). This notion of generativity in digital infrastructures suggests that embeddedness in an *a priori* organisational context and existing installed base is of diminishing importance to digital infrastructures, to a point where a pre-existing organisation might not be necessary for subsequent organising by online communities or others. Instead, organising (notably by online communities) may emerge independently or in the periphery of a pre-existing organisation. As flexible digital infrastructures and organisation co-evolve (Tilson et al. 2010), the question therefore becomes one of how self-organising takes place in online communities based on digital infrastructures without a pre-existing organisation. As members of OS communities, a subset of online community, are the most in command of the digital infrastructures that mould their community, they make an ideal community for study. This paper therefore seeks to answer the research question: *What is the role of code forking in digital infrastructures in the self-organisation of OS communities?*

We address this question through a multi-method, longitudinal case study of the emergence and evolution of the Bitcoin community from the Blockchain infrastructure over the course of six years. The Bitcoin

[1] Although there is some discussion around how much control is required to retrospectively change the blockchain, see e.g. Eyal, I. and Sirer, E.G., 2014, March. Majority is not enough: Bitcoin mining is vulnerable. In *International Conference on Financial Cryptography and Data Security* (pp. 436-454). Springer Berlin Heidelberg.

Blockchain, while an infrastructure, does not have an organisational legacy. Instead, the infrastructure was created in isolation: by an anonymous individual or group of individuals who later severed ties with the project. As such, Bitcoin does not have an *ex ante* organising base, as it was severed from whatever organisational base it once had. Instead, organising emerged from and around the infrastructure as it evolved. For the time being, the Bitcoin Blockchain is an extreme and isolated case. However, it is instructive in providing insight into the larger phenomena of organising associated with digital infrastructures (Siggelkow 2007), especially as digital infrastructures gain significance within both existing and emerging organisations. It is worth noting that the Bitcoin Blockchain has inspired a variety of other Blockchain-based infrastructures (e.g. Ethereum, hyperledger, etc.), but for the sake of being able to draw clear empirical boundaries we focus only on the Bitcoin Blockchain. This research may nevertheless have implications for understanding these second (and third) generation(s) of digital infrastructures.

This paper is structured as follows. First, we examine the existing literature on digital infrastructures to identify organising principles. Based on our review of existing literature, we propose and substantiate how coding, and specifically code forking, acts as a mode of organising. Thereafter we present the case background and research design, before showing how code forking in the blockchain infrastructure led to organisational change in the Bitcoin community. Finally, our research findings and their implications for –organising among online communities are discussed.

Organising in Digital Infrastructures

Star and Ruhleder (1996) observe in their seminal work on infrastructures that infrastructures are most apparent when they fail. This ability of infrastructures to recede into the background highlights how they emerge: someone does something in relation to someone (or something) else, thereby mobilising a collection of artefacts, which we consider emergent digital infrastructure, e.g. a customer buys goods from a vendor via a card payment infrastructure, colleagues exchanging emails etc. Infrastructure therefore relies on some form of organised activity by an group of actors in order to be considered an infrastructure: "Analytically, infrastructure appears only as a relational property, not as a thing stripped of use" (Star and Ruhleder 1996, p. 113).

Consequently, digital infrastructures have been described as a as the layer of code upon which both platforms and applications are built (Tilson et al. 2010). Infrastructures entail the employment of technology to facilitate existing organisational practices (Vaast and Walsham 2009). Henfridsson and Bygstad (2013) observe that infrastructure is often used as an independent variable to support other organisational processes and aims, for instance knowledge creation (Malhotra et al. 2004), performance gains (Rai et al. 2006) or outsourcing processes (Tanriverdi et al. 2007). Such organising practices are a foundational element of infrastructure: an infrastructure is fundamentally something, which supports some organised relational practice through which it is actualised (Star 1999). The focus of previous theorising around digital infrastructure has emphasised how organisational structures (Henfridsson and Bygstad 2013) and boundary objects (Eaton et al. 2015) determine infrastructure development.

A larger discussion in the information systems literature calls for research to interrogate the importance of digital artefacts in their own use and perpetuation (Leonardi 2013; Orlikowski and Robey 1991), especially in organisations and organising processes. However, in the context of infrastructures, this call has remained unanswered. This may largely be because it is hard to conceive of the development of an infrastructure without an overarching set of organisational aims in mind.

Existing Views on Infrastructure Organising

Digital infrastructures have been argued to possess an "...overall capacity to produce unprompted change driven by large, varied, and uncoordinated audiences" (Zittrain 2006, p. 1980). The result is emergent (Edwards et al. 2007) and even 'accidental' progress, or innovation (Austin et al. 2011). However, previous studies have not explored the mechanisms that explain how digital infrastructures, apart from those wholly controlled by single organisations (Ciborra 2000; Eaton et al. 2015), lead to this progression or innovation through organising. This begs the question of whether digital infrastructures, in an age of increased automation and digital ubiquity, do indeed carry the seeds of their own evolution.

Received literature on digital infrastructure represents at least four distinct principles representing different views of how organised social practices give rise to digital infrastructures; namely through adaptation, inscription, interaction, and management choice.

First, adaptation views see digital infrastructure evolution as a result of the efforts of distributed human actors to adapt to their environment and to other actors (Braa et al. 2007; Hanseth et al. 2006). Adaptation views build on developments of complexity theory (Holland 1995; Mol and Law 2002). For example, Hanseth & Lyytinen (2010) propose a theory of how digital infrastructure design can account for the adaptability of distributed actors, and Nan (2011) explores how the use of distributed digital technology emerges as adaptations between users, technology and tasks.

Second, inscription views describe how infrastructures evolve as human actors translate and inscribe their interests into assemblages of technological components, thereby seeing infrastructures as evolving networks of human and non-human actors (Aanestad and Jensen 2011; Hanseth and Monteiro 1997; Yoo et al. 2005). Building on actor network theory (Callon 1986; Latour 1987), inscription views emphasise the relationship between human actors and technology in translating and inscribing behaviour in structural terms and see infrastructure evolution as changes to a set of relations between humans and technology as human actors mobilise resources to support some more or less organised action. For example, Eaton et al. (Eaton et al. 2015) describe how the tuning of boundary resources by a network of distributed human actors affect digital infrastructure evolution.

Third, interaction views argue based on the premise that an infrastructure's evolution should be seen as a process of continuous interaction between its users and stakeholders as they engage in sensemaking around an organised activity. Drawing from theories of collective learning and work practices (Lave and Wenger 1991; Wenger 2000), interaction views see infrastructure evolution as a result of interactions within a given community-of-practice resulting in the formation of socio-technical relations (Pipek and Wulf 2009; Star and Ruhleder 1996; Vaast and Walsham 2009). For example, sustained participants in OS communities consistently engage in situated learning that both made conceptual contributions of advising others and practical contributions by improving the code (Fang and Neufeld 2009).

Finally, management views emphasise the role of management decisions in facilitating infrastructure evolution. Infrastructure evolution is seen as a process by which managers initiate and implement changes to information technology infrastructure in order to align strategic IT capabilities and strategic objectives (Beckert 1999; Child 1997). For instance, Broadbent & Weill (1997) explain how managers through thorough understanding of the strategic context of their organisation can define maxims to determine the infrastructure capabilities they should implement to achieve their business objectives.

Table 1. Organising through digital infrastructure

Organising principle	*Description*	*Theoretical foundation*	*Example references*
Adaptation	Distributed actors adapt to their environment through changes in tasks, technology and relations	Complexity theory	Hanseth & Lyytinen (2010) Nan (2011)
Inscription	Existing organisational practices are inscribed in technological artefacts	Actor Network Theory	Aanestad & Jensen (2011) Eaton et al. (2015) Yoo et al. (2005)
Interaction	Interactions in a community of practice resulting in new socio-technical relations	Collective learning and communities-of-practice	Fang & Neufeld (2009) Pipek & Wulf (2009)
Choice	Choice of infrastructure governance and organising as a result of informed management decision	Strategic choice theory	Beckert (1999) Broadbent & Weill (1997) Child (1997)

The principles outlined in Table 1 all share three main assumptions about infrastructure organising: first, that infrastructure-organisations are built upon pre-existing organised practices (Star and Ruhleder 1996). Second, in the course of infrastructure evolution, human behaviour is inscribed into the technological components of the infrastructure (Hanseth and Monteiro 1997), and third that changes to the infrastructure require coordination among heterogeneous and distributed human actors (Ciborra 2000; Hanseth and Lyytinen 2010).

However, these assumptions mean that existing conceptions of digital infrastructures do not fully account for the emergence of the Bitcoin infrastructure. Moreover, they fail to account for a situation in which human behaviour is constrained by the infrastructure itself, such that adoption becomes a key organising principle. In the following, we propose and substantiate an additional organising mechanism of digital infrastructures: adoption through code forking.

Coding as Organising

Infrastructures have been said to evolve based on common organising conventions, and rely on installed base inertia in adhering to shared standards (Edwards et al. 2007; Star and Ruhleder 1996). One of the areas in which maintenance and changes in digital infrastructures are visible is at the level of the source code which comprises the infrastructure (Nyman and Lindman 2013). Indeed, source code is the very material that dictates how the infrastructure works, including the rules whereby platforms, applications and other modules can connect with it.

As research attention turns to the importance of digital materiality in organising (Gherardi 2009; Leonardi 2011; Orlikowski 2007), the practice of coding of digital infrastructures is increasingly deserving of consideration when it comes to its role in organising. Indeed, although some conceptions of materiality consider only those things with tactile embodiment as "material" (Orlikowski, 2007), others argue that all "objects, sites, and bodies" (Ashcraft et al. 2009, p. 2) that have significance should be considered in organising (Leonardi, 2010). Without engaging in the debate around whether digital code should be considered material, we nevertheless propose that insofar as such code affects social and organising processes it should be considered as a mechanism of digital infrastructure organising.

The complexity of digital infrastructures at large-scale goes beyond that of traditional systems design (Hanseth and Lyytinen 2010; Henfridsson and Bygstad 2013; Tilson et al. 2010; Yoo et al. 2010). Due to the scale and complexity of digital infrastructures, distributed forms of control are often the only way to organise digital infrastructure (Hanseth and Lyytinen 2010; Star and Ruhleder 1996). In general, the sheer task of maintaining the infrastructure requires more resources and knowledge than a single person or organisation possesses, leading to a distribution of both control and decision-making structures (Yoo et al. 2010). This means that digital infrastructure effectively become 'doubly distributed' networks in which "…both organizational and technological controls are distributed among heterogeneous actors and artifacts " (Yoo et al. 2008, p. 1).

The heterogeneity and programmability of digital technologies have led to new forms of generative and distributed organisations, where digital technologies and organising meld together (Dhanarag and Parkhe 2006; Yoo et al. 2008). In other words, the creation of organisational form takes place through the production of computer code. Digital code has dynamic (Aho and Hopcroft 1974; Kitagaki and Hikita 2007) and even agentic capabilities (Andersen et al. 2016). The importance of digital code for infrastructure organising therefore lies not only in how it represents organisational practices, but also in how it plays an active role in organising.

Building on existing research on digital infrastructure (Hanseth and Lyytinen 2010; Henfridsson and Bygstad 2013; Tilson et al. 2010) we therefore propose that digital infrastructures, through generative and emergent changes to digital code, have the potential to foster new forms of organising. This is particularly through the adoption of code. Next, we move on to consider in greater depth the ways in which organising by OS communities emerge in the absence of organisational embeddedness of digital infrastructures.

Code Forking as Self-Organising

The most well-understood organisation-first communities of source code developers are OS developers. These developers initiate and organise themselves around the desire to find a solution to a particular problem, or 'shared itch' (Raymond 1999), and produce source code in order to do so. In these communities, both changing and maintaining the source code (infrastructure) is done jointly, and both bugs within the code, and threats to the infrastructure (for instance from hacking) are dealt with collectively by members of the community. Among such communities, changes to the underlying code are commonplace, and expected (Fang and Neufeld 2009). Often there is consensus as to what should be changed or fine-tuned, and why. Such changes to the code are discussed among developers and contributors and, as such, visible in, for instance, online forums (Phang et al. 2014), although it may take negotiation to come to an agreement and some members of the community may be more active than others (Phang et al. 2015). These projects are run against the backdrop of an OS licence. Although there are many kinds of OS licence, they typically allow, at a minimum, the free re-use of code covered by that licence. As a result, splits from the original OS project cannot be prohibited, although are typically discouraged (Nyman 2015).

Given the fact that maintenance of the infrastructure—and therefore its evolution—is shared, what happens when there is a disagreement about the future of the infrastructure from within the community? In non-distributed and proprietary settings, the problem is solved with reference to the organisation's hierarchy (Dahlander and Magnusson 2005; Kartseva et al. 2010). However, in the case of an OS community, the answer is less simple: in general, members of the community try to come to a negotiated settlement (Nyman 2015), but in exceptional cases some developers diverge in their opinions of the future of the project, and two (or more) inconsistent versions of a project are created.

These spin-outs are known as 'forks', and are defined as when "a part of a development community (or a third party not related to the project) starts a completely independent line of development based on the source code basis of the project" (Robles and González-Barahona 2012, p. 3). These have been classified as having three types: code fragmentation, pseudo-forking, and code forking (Raymond 1999). The first two types involve the distribution of the original code along new channels, but the resulting new distributions of the code are both compatible with the old version and benefit from future developments in the parent code (Nyman 2015). However, a code fork is a complete change in the underlying code such that the new version of the code and the old version of the code are forward incompatible. In previous studies of forks, only one kind of fork has been observed. However, where source code is used to support infrastructure, both forward and backward compatibility are at issue. This is because an infrastructure can contain a historical record in a way that other OS projects may not need to. We therefore distinguish between these two kinds of forks; namely the "soft" fork, which is only forward-incompatible, and the "hard" fork, which creates a fork that is both forward and backward incompatible.

These forks are typically frowned upon by the OS community, largely because of the impact on both community and individual developers' reputations (Nyman 2015; Weber 2004) and because multiple, incompatible versions of a software can discourage related future developments (Meeker 2008; Nyman 2015). Consequently, within and beyond OS communities, forks in the source code are a key visible element in the instantiation of digital infrastructure organisation. As such, the notion of code forking provides a theoretical lens through which digital infrastructure evolution can be studied. What follows is a description of how we apply the notion of forking in a longitudinal research design.

Research Design

In order to answer the research question of how digital infrastructures evolve new forms of organising in OS communities, we conducted a longitudinal multi-method (Venkatesh et al. 2013) study of an emerging digital infrastructure covering a period of six years. In the following section, we first discuss our case selection and background before explicating data collection and analysis.

Our choice of case was driven by the need to meet four basic requirements: first, the infrastructure had to be exclusively self-organising in the sense that the organisation around it would have emerged as a consequence of specific technological changes. Second, we had to be able to identify distinct instances of forking and adoption. Third, the infrastructure selected as case setting needed a history spanning over a longer period of time allowing us to study its evolution in both detail and scale. Finally, the infrastructure should have good records of both code forking and organising practices to allow us to analyse the implications of code forking on organising.

To our knowledge, there is currently only one infrastructure that meets these requirements: the Bitcoin Blockchain is self-organising, relatively long lived, and has good digital trace records. Although the OS software has since been re-used to create new infrastructures (e.g. Ethereum, Ripple), these next generation applications are still emergent and, as cases, are still ongoing and therefore trickier to study (Yin 2003).Moreover, as an infrastructure with a sizeable and distributed user and supporter base, forum data provides good records of both when forks occurred (or could have occurred) and the underlying social contexts of these forks.

Data Collection and Analysis

Our aims with this research were 1) to identify and understand code forking events of significance to the organisation of the Bitcoin community as well as their broader context, and 2) to establish the role of code forking on the self-organisation of the Bitcoin community. Our data collection and analysis therefore took into account these two objectives.

Table 2. Overview of data collection and analysis

Data source	*Description*	*Analytical techniques*	*Research outcome*
Digital traces	314 551 digital trace records of interactions collected from the bitcointalk.org community over a six-year period, including records referring to the Bitcoin source code as well as organisational changes	Latent Dirichlet Allocation (LDA), the results of which were coded to identify the main forking events that occurred, and the context in which they occurred	Identification and characterisation of organisational changes during each forking event
Interviews	10 formal, semi-structured interviews recorded and transcribed verbatim. The interviews focussed on key events in the Bitcoin community (forks, regulatory changes, stigma), as well as on the interviewees' understandings of the community and infrastructure's strengths and weaknesses	We used open and axial coding to produce analytical memos (Miles and Huberman 1994) around the significant code forks, understand the background of the community, and highlight major associated events. This enabled us to verify findings in the primary data source, namely the forum data.	Identification of a sequence of significant events in which the Bitcoin source code was forked, including the type of forking for each event.
Documents	56 Press articles 71 Blog posts on topics related to forks and other conflicts (e.g. political ideologies) in the community from other sites (e.g. Bitcoinfoundation.org, Coindesk.com, Techcrunch.com, medium.com)		Identification of environmental conditions and important periods in the history of the Bitcoin community

Overall, our methods were grounded in inductive reasoning and rested on the use of three sources of data: a series of ten interviews with Bitcoin entrepreneurs, digital trace data from the Bitcoin online community, and extensive documentation and (see table 2). Our primary data source was forum data, and our computational analysis was triangulated against interview data and documentation. This was in order to 1) imbue our computational findings with context, as provided by interviews (Gaskin et al. 2014) , and 2) to ensure the veracity of our findings.

We collected the digital trace records (Hedman et al. 2013; Howison et al. 2011) of community interactions across 314 551 interactions scraped from the Bitcoin forum bitcointalk.org covering a period from October 2010 to June 2016. Bitcointalk.org is a forum dedicated to discussions around Bitcoin, primarily in English. It is among the most prominent forums used by Bitcoin enthusiasts. However, unlike mainstream forums like Reddit.com, it is often used specifically by Bitcoin professionals meaning that interactions on Bitcointalk.org are particularly linked to the development of the Bitcoin code base. Furthermore, it contains sections that are both general and specific in nature; for instance, threads around the technicalities of the Blockchain and mining, as well as more discussions of a more organisational nature. We opted to examine forum data rather than a code repository like Github as the motivations and context of technological change, as well as the resulting organisational practices, are better reflected in this kind of fine-grained semantic data.

We began by analysing forum data from the Bitcoin community across five phases. Each period was analysed separately using computational techniques on digital trace data (Hedman et al. 2013; Howison et al. 2011) to generate open codes for each time interval, analogous to what is done in manual coding (Glaser and Strauss 1999). The collected digital trace data was divided up into five phases: October 2010-December 2011 (57 220 interactions), January-December 2012 (84 100 interactions), January 2014-July 2015 (122 409 interactions), August-September 2015 (25 431 interactions), and October 2015-June 2016 (25 379 interactions). We began a first level coding of the data using the computational natural language processing technique Latent Dirichlet allocation (LDA) implemented in the open source statistical software R (Blei et al. 2003). LDA is a generative topic model that reveals patterns in a set of documents by extracting unobserved groupings (latent themes) based on semantic similarities between different parts of the data (Sievert and Shirley 2014). LDA discovers latent themes within a collection of documents by sampling a topic for each word at every iteration of the algorithm and ranking words based on their relevance to each topic, which therefore has a unique distribution over words that can be compared using cosine similarity measures (Chuang et al. 2012).

Analysing semantic clusters of terms by topic allowed us to discern combinations of topics under discussion by users related to each forking event. This period-by period clustering was then compared with analytical memos (Miles and Huberman 1994) generated through an analysis of 10 interviews and a number of online documents to ensure its validity. These additional sources were necessary because of the risk of losing context when conducting computational analyses (Gaskin et al. 2014). Thus, while forums provided the primary data (and the main analytical findings), analytical memos based on other data sources provided vital context. These analytical memos were created after open and axial coding of the interviews and documents described in Table 2 (Glaser and Strauss 1999). During coding, we looked for important forking events and the type of forking involved, as well as for the social environmental contexts co that influenced the Bitcoin community.

The three sources of data were combined to generate a thick longitudinal analysis of the organisational antecedents and outcomes of specific code forks in each period. Based on this analysis, we produced a thick description (Bechky 2006) of the role of code forking, as a mechanism, in organising in the Bitcoin community.

Patterns of Self-Organising in the Bitcoin Community

Our longitudinal analysis of the Bitcoin infrastructure and community revealed three distinct patterns by which the practice of developing the underlying source code was instrumental in shaping how not only the digital infrastructure but also in (self-)organising the online community. These patterns of potential self-organisation become visible on the level of code forks, with adoption turning these potential changes into actual changes.

What is interesting to note is that code forks occurred in response to the changing environmental conditions in which the infrastructure existed. This is consistent with existing understandings of infrastructure as including organisational practices and contexts, as well as the artefacts themselves (Star and Ruhleder 1996). As this was a longitudinal study, we could examine not only when and why the code forking in the digital infrastructure evolved into potential new organisational forms, but also when new ideas were incorporated into the existing infrastructure through adoption. The three distinct forms of code forking i.e. development forks, pseudo-forks and hard forks, their environmental conditions, and an overview of the sequence of forking events of importance to the online community are contained in Figure 1.

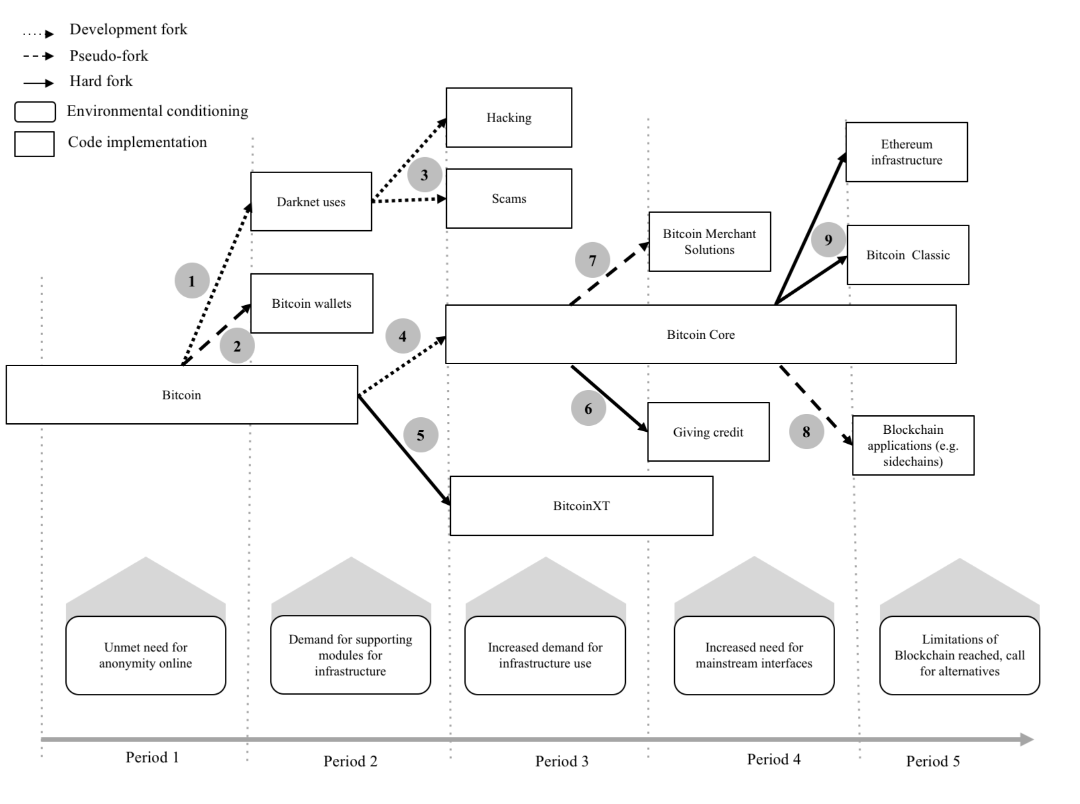
Development fork
Pseudo-fork
Hard fork
Environmental conditioning
Code implementation
Bitcoin
Darknet uses
Bitcoin wallets
Hacking
Scams
Bitcoin Core
BitcoinXT
Bitcoin Merchant Solutions
Giving credit
Ethereum infrastructure
Bitcoin Classic
Blockchain applications (e.g. sidechains)
1
2
3
4
5
6
7
8
9
Unmet need for anonymity online
Demand for supporting modules for infrastructure
Increased demand for infrastructure use
Increased need for mainstream interfaces
Limitations of Blockchain reached, call for alternatives
Period 1
Period 2
Period 3
Period 4
Period 5

Figure 1. Forking of the Bitcoin infrastructure

In the following, we analyse how each form of code-forking led to a specific pattern of self-organising. Inspired by terminology well-established in systems biology literature, we address how different types of self-organisation occurs in response to environmental stimuli (Zimmer and Emlen 2013).We have called these patterns of self-organisation 'speciation', 'variation' and 'adaptation'. Each pattern of self-organisation is illustrated through an empirical example and related to similar instances in order to discern the specific characteristics of each pattern.

Hard Forks as Organisational Speciation

The first pattern of self-organisation is driven by a hard fork to the underlying code; that is, a fork in the code that created a new infrastructure that is both forwards- and backwards- incompatible with the existing infrastructure. If the community adopts this 'forked' version of the code *en masse*, it will become the dominant infrastructure. In fact, one might even say that it becomes the infrastructure, while the previous versions of the code are discarded or held by a less influential minority. We will refer to this pattern as 'speciation', or the creation of a new species of infrastructure

A particularly illustrative example of a hard fork is that of the BitcoinXT (fork 5). In June 2015 two prominent developers of the Bitcoin source code suggested that the sustainability of the project was in jeopardy. In late 2015, Bitcoin transactions began to be delayed and a backlog of transactions grew. In other words, the infrastructure began to fail (Hearn 2015). An infrastructural shift, known as BitcoinXT ("XT"), necessitating a change in the underlying source code was proposed as a solution. This proposed solution would entail increasing the size of each block in the blockchain from 1mb to 4mb:

> *As Bitcoin has grown, so have the blocks. Reasonable traffic projections indicate that as Bitcoin spreads via word of mouth, we will reach the limit of the current system [with a 1mb block size] sometime next year, or by 2017 at the absolute latest. And another bubble or press cycle could push us over the limit before even that. The result might not be pretty. So it is now time to raise the [block size] limit, or remove it entirely.* - Mike Hearn, Aug 15, 2015

Increasing the block size would reduce the number of miners able to run the software (owing to issues around processing power), but would increase Bitcoin's transaction handling capacity. Opponents of this change labelled the original version of the infrastructure Bitcoin Core, and they argued that, among other things, XT was untried and may not scale well (with which XT proponents disagreed). They also argued that this change would make the project more centralised, putting more power in the hands of fewer miners—who could make more drastic changes in the future. This back-and-forth shows how political beliefs affected how code changes were perceived, affecting willingness to adopt. In essence, they agreed with the need to do something, but argued that the technical shift to XT would change the social practices whereby the infrastructure operated, namely by centralising control of the source code. OS communities often identify strongly, so controversial attempts to change it are often taken personally (Ren et al. 2012):

> *Bitcoin's own former lead dev, Gavin, and his henchman Hearn are in the process of sabotaging Bitcoin from the inside. You will hear about it when their XT Trojan horse deploys its payload, and attempts to force us all to join their altcoin at Bitcoin's expense.* - July 9, 2015, 03:42:23 PM

The rules for how a potential change to the root code of the Blockchain can occur are enshrined in the source code. In essence, minor changes that create compatible versions of the software are dealt easily, while major changes, like in this "fork", require active adoption. Only majority adoption of a change will mean that the root source code is changed. Participants in the network, whether miners or entrepreneurs running platforms on the Blockchain infrastructure, have to choose which version to run. Ultimately, the version that garnered the most support would become the "real" Blockchain.

The attempt to fork the Blockchain source code ultimately resulted in two forward- and backward-incompatible versions of the Blockchain: Miners did not adopt the XT version of the code in sufficient numbers to make it the dominant version of the Blockchain. Moreover, swathes of the Bitcoin community also boycotted entrepreneurs and users who switched over to XT. The changes to the source code were therefore not internalised, resulting in what would become what the community called a "hard fork" to the source code. Unlike a soft fork, wherein the new version of the infrastructure is compatible with those running the old version of the software, a hard fork would entail those running old versions of the software being unable to read/recognise changes in the new version.

This hard fork to the Blockchain was prompted by environmental changes around the infrastructure, to which a response was needed. The source code governed what changes could and could not be made, and the fork itself became a reality through two changes. First, through code changes to the infrastructure, and second, through adoption. Substantial adoption above a certain threshold was needed for the material, code-level changes in order for the new version to be considered the "real" infrastructure, especially as the changed source code was prospectively and retrospectively inconsistent with the previous source code.

This hard fork created two 'species' of infrastructure; the original Blockchain and the alternative Blockchain that resulted from a hard fork that substantially altered the underlying source code so much that it made the two pieces of infrastructure inconsistent with one another. However, this hard fork was just one instantiation of evolution. In order to test for the generalisability of a hard fork leading to speciation in infrastructure evolution, as well as to test for other possible mechanism of digital-first infrastructure evolution, we expanded our observations and inductive analysis to span 6 years. At the time of data collection, this was the lifetime of the Bitcointalk.org forum dedicated to the infrastructure and the community.

Other instances of speciation through hard forks include fork 7 in which some members of the Bitcoin community saw the future of the infrastructure as being as a part of the existing financial system, not as a parallel currency system. One of the sticking points in this area was the fact that more Bitcoins could not be created at will; the Bitcoin source code dictated that they could only be created through mining, and at a decreasing rate up to a maximum number. However, the mainstream financial system relies heavily on credit. In early 2013, this was seen as both a merit of the Blockchain system and a possible stumbling block to greater integration of the mainstream and Bitcoin financial systems. Altering the system such that it could issue credit would have required a substantial change to the Blockchain system. Unlike the XT change, the environmental pressure exerted in favour of this possibility was not sufficient to drive the hard fork it would take to make it a reality. As such, it was a hypothetical hard fork that was never adopted at the level required to become a new form of organising.

The final hard fork (fork 8) revolved around alternative uses of Blockchain-like infrastructures other than for Bitcoin, and is driven at least partly by the limitations the Blockchain was seen to have at that point. Users point both to the fact that Bitcoin transactions are slow (as was the case with XT), and to the fact that the Blockchain in its then-incarnation did not easily allow for the storage of other kinds of information, or the execution of smart contracts. This results in several major hard forks, the most-discussed of these are Bitcoin Classic and Ethereum. Bitcoin Classic, like XT before it, proposes to increase the size of the Blockchain in order to facilitate a larger number of transactions more quickly, while Ethereum is developed to facilitate the use of the Blockchain technology for other kinds of applications, for instance smart contracts, and has effectively resulted in the emergence of a separate OS community. This community is both operationally and ideologically distinct from the parent community.

These three different hard forks represent substantial organisational changes. In the case of this distributed infrastructure, the code forking mechanisms around adoption require that substantial changes be backed by substantial adoption. In essence, the code supports and reinforces the status quo, while changes to it would require significant resources and mobilisation on the part of would-be forkers. The self-organisation mechanisms at work here could pan out in one of two ways: for the hard fork to be absorbed by the infrastructure would require adoption by users on a large scale. There is therefore a social element to an otherwise technical evolution; the combination of the two are what leads to internal evolution, or what we later call “variation”.

The exact implications of organising depend on the precise changes made to the underlying source code, but in the case of hard forks, change necessarily involve completely new organisational structures. In the case of a shift to XT, for instance, the technical centralisation of the infrastructure (by making mining harder and therefore raising barriers to entry) would put control over the infrastructure into the hands of fewer—meaning that future decision making would be controlled by a sub-set of the existing (or a new) community. Moreover, as a larger block size would facilitate wider use of the Blockchain and make it a competitor for the likes of Visa and Mastercard, its mainstream appeal is likely to change the composition of the community for which Bitcoin has become known, partly through the change in control structures and partly through changing who has an interest in maintaining the infrastructure. Thus, this fork creates new organisational forms in the community: it changes power dynamics and thus future decision-making, as well as the composition of those in power.

Thus, the creation of a whole new organisation—with new practices and meanings—requires both a change in source code and a substantial amount of adoption. Speciation is therefore an instance of infrastructure evolution framed according to code-level changes, and implemented at the level of user adoption.

Development Forks as Organisational Adaptation

Variations on the level of code use were not the only forms of self-organisation driven by forking of the underlying source code. Adaptation, or developments that added to the underlying source code, were also important and influential drivers of community organising. Development forks built upon the underlying source code to enable new technical functions which, in turn, led to adaptation in the community through a combination of code and practice changes.

One of the most extensive, and common, adaptations revolved around how the infrastructure led to the development of entrepreneurship-focused adaptations of Bitcoin (fork 2). These adaptations involved the development of code that built upon the existing source code without introducing incompatible elements. Some examples of this include the development of wallets to hold Bitcoin currency, and analytical tools that conduct analytics on the contents of the underlying blockchain infrastructure.

As these developments built upon the existing source code, they enabled new code uses; most notably entrepreneurship through the creation of start-ups and corporate ventures. However, they also incorporated the organising elements inherent in the underlying source code. One member of the community expresses how new adaptations were accepted by the community under certain conditions:

> *Please understand that I have great respect for the work you've done. Your service is very well constructed and well-loved for good reason. While some of the things I've brought up might be improved with some tweaks here and there, much of it is simply the structural consequences of centralized services, trusted parties, web clients, etc...I don't think our community should take any actions which promote centralization or consolidation due to systemic risk if nothing else... I don't believe it should promote your wallet service either.* - December 03, 2012, 03:02:53 PM

But entrepreneurship did not just extend to services that catered to the existing community. Instead, some of the adaptations facilitated by the infrastructure allowed the infrastructure to interface with other infrastructures, enabling inter-organisational linkages. One example is that of a Point-of-Sale (PoS) adaptation (fork 6), which built both upon the underlying infrastructure and on other adaptations to extend the usefulness of the infrastructure:

> *We have released an update of our Pos [Point-of-Sale] software witch includes a module to connect and process payments trough Bitcoin-Qt wallet. So with this update, more than 3000 local busineses over the world who now are using Sysme Pos as point of sale software can have the tool to accept Bitcoins We hope that this will encourage them to accept Bitcoin as payment so this this project can make a step further.* - June 05, 2013, 09:53:11 PM

Such an addition was not only useful on the level of use, it also served to bring more stakeholders in when it came to supporting and engaging with the infrastructure. While the creation of entirely new adaptations led to the creation of new child organisations, adaptations that led to linkages with existing organisations had different organisational implications. They facilitated the transfer of some of the infrastructure's organising practices, in whole or in part, to other hitherto unaffected organisations.

Among the adaptations that built upon the infrastructure are those that would allow, in an indirect way, the infrastructure to perform additional functions. While the Blockchain's original architecture was intended as a proof-of-concept for the transfer of a currency, one adaptation allowed for the transfer and maintenance of a centralised database of something other than that currency (fork 9). In an adaptation known as a sidechain, developers built upon the Blockchain to allow for an object-agnostic transfer which interfaced with the Blockchain:

> *The paper proposes two-way pegged sidechains as an extension mechanism for Bitcoin. The idea is that coins would be able to move from the Blockchain, to a sidechain, and then back again in a trustless way. This would allow sidechains to implement properties that are not feasible to implement on Bitcoin itself, while preserving the total number of Bitcoins.* - September 10, 2015, 03:44:19 PM

Though it represents a drastic development, this fork was still consistent with the underlying source code. This meant that, like in forks 2 and 6, the new organisation that formed around the adaptation also relied on elements of the infrastructure's coded-in organising practices in order to function. Thus, organisational adaptation was enabled (and constrained) by source code, through development forks. These development forks added to the existing organisation by attracting new users to the community, and by making the infrastructure itself able to support more things—thus changing its character on the code level.

Pseudo-Forks as Organisational Variation

One of the most compelling patterns of self-organising that emerged from our longitudinal examination was how the flexibility of the Blockchain source code not only drove substantial changes in the underlying source code, but also permitted organisational variation when the environment encouraged them. These instances of variation are visible in forks 1, 3 and 4, with the first two relating largely to the use of Bitcoin for illicit purposes, and the third related to the identification of the existing Blockchain as 'Core'.

During the early stages of the evolution of Bitcoin in 2011 and 2012 (forks 1 and 3), many different kinds of users saw the new infrastructure as a potential for new ways of conducting transactions outside the established financial system. These forks, while seemingly something out of the ordinary do not entail any underlying code change. They are therefore called 'pseudo forks' and entail a variation in the patterns of code use—but nevertheless ones that have organisational implications. These new use patterns attracted a number of people whom used the infrastructure to conduct illicit activities. As one user observes:

> *...I saw many a lot of devious schemes to earn bitcoin. Just like HYIP, PONZI, Gambling, Bet etc. I am very afraid of RIBA. therefore, for guidance and assistance will be appretiated. Thanks* – March 03, 2012, 12:38:21 AM

This was made possible by the nature of the source code itself. Indeed, the qualities of decentralisation and semi-anonymity—seen as elements of building a new technical system based on trust in a system rather than individuals or institutions—paradoxically also gave users the possibility to use the system for transactions that many in the community considered to be unethical, including transactions involving drugs on the Darknet,[2] and Bitcoin-denominated scams:

> *Of course, bitcoin is not the problem. People who misuse the bitcoin and abuse is are the ones who are the problem. If terorrist are using bitcoin, not really nice of them to shed bitcoin in a bad light.* - November 24, 2015, 02:36:19

Another pattern of variation occurred when the Blockchain began to experience difficulties processing transactions: community members began discussing alternatives and changes to Blockchain to deal with these transaction lags (fork 4). Defenders of the existing infrastructure, including its existing coded-in form, began to emphasise the organisational implications of the existing infrastructure. Moreover, they labelled the existing infrastructure 'Core' in response to attempts to change the source code. Much like the variations in practice that use of the infrastructure for illicit transactions, this kind of fork is an instance of practice-level variation on how the infrastructure is used:

> *The code which powers the Bitcoin network can be found here: https://github.com/bitcoin/bitcoin. This code has evolved as long as Bitcoin has been around. But the debate over the block size limit and how to manage it has caused some of the best-known developers to set up another client for bitcoin, here: https://github.com/bitcoinxt/bitcoinxt... The former is now referred to as "Bitcoin Core", and the latter "Bitcoin XT". ... From the XT github README: "Bitcoin XT is more experimental than Bitcoin Core, and has a strong emphasis on supporting the needs of app developers and merchants. By running it you not only provide additional services to the network but help build confidence in the implementations, contributing towards consensus for inclusion in a future version of Bitcoin Core.* - May 31, 2015, 05:28:36 PM

These new patterns of code use arose both in response to the possibilities that the infrastructure presented, as well as the environment in which both actors and the community found themselves. In this case, organisational structures are changed through the proliferation of multiple ideologies, and the polarisation of certain practices within the community. Moreover, these practices may attract additional community members, affecting how the community interacts.

Source code, through pseudo-forks, therefore led to organisational variation. While interpretation and narrative changed the structure and dynamics of the organisation, the pre-existing code limited (and enabled) the possible interpretations. Having demonstrated how code forks enable and constrain organising, and the subsequent role of use/code adoption, we turn now to discussing these findings and their implications.

[2] The most prominent case is that of Silk Road, an infamous online drug market that makes extensive use of the cryptocurrency Bitcoin.

Discussion

Code forking involves both changes to the source code itself, and to the interpretation and use of source code for new purposes. Our findings show that different variations in these two parameters lead to different organisational outcomes. The configurations of changes to source code and code use amount to distinct patterns of self-organising. The following discusses each of these patterns of self-organising that emerge through code forking and adoption practices.

Speciation is the process by which new organisational structures emerge from radical breaks in the source code. It requires both changes in the underlying code and in its adoption. For speciation to occur, code changes must replace existing source code base, whereby it creates a new infrastructure that is technically incompatible with the existing source code. This requires a complete reworking of the way in which the replacement code is produced, maintained, and applied in various contexts. This results in the creation of new organisational structures with new organising logics. In the case of Bitcoin, hard forks to BitcoinXT or credit based instantiations of the infrastructure would both have resulted in the introduction of a trusted third party acting as a middle man between users. This would effectively introduce a new organisational structure departing from the distributed, non-hierarchical organisation of the original infrastructure.

Adaptation refers to the process by which new supplementary organisational structures emerge by leveraging the infrastructure's existing source code. Adaptation can be seen to add to existing organisational structures by adding to the underlying code, and supplementing community membership. Adaptation involves code changes that rely upon the existing source code base and add to it, resulting in new use domains. The resulting supplementary organisational structures represent additions of new, yet compatible, organisational structures operating within the existing organising logic of the infrastructure. Examples of this in the Bitcoin infrastructure include the introduction of new entrepreneurial products based on the core infrastructure whereby a new company is formed, which is reliant on the underlying infrastructure, and in some cases on other supplementary organisations.

Variation refers to the process by which changes in the area of application of existing code, or in interpretations of the existing code, can connect seemingly unrelated organisational structures to that of the infrastructure. Variations in the purpose of existing source code does not require any actual changes to the source code. Instead, the source code enables hitherto unexpected practices, which themselves have organisational implications. Examples from the Bitcoin case include fraudulent and illegitimate applications such as outright Ponzi schemes and gambling applications designed to bypass existing regulation to allow for higher stakes even in high-risk games.

The three types of code forking, how they manifest in code implementation and use as well as their antecedents and consequences are shown in figure 2.

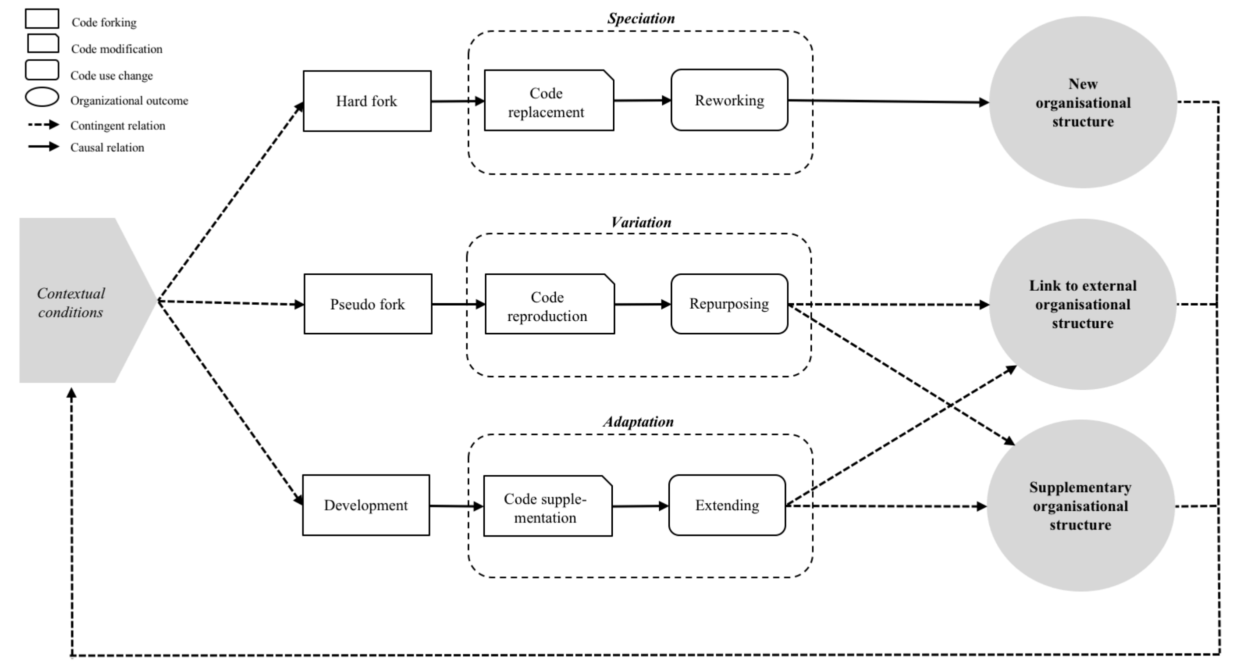

Figure 2. Patterns of self-organising through code forking

The changes to source code and code use outlined in figure 2 are relational in nature and the maintenance of the community infrastructure necessitates constant interaction. Digital infrastructures, unlike platforms and applications, do not have strict boundaries and cannot be defined through a specific set of functions or modules (Tilson et al. 2010). Instead, their boundaries are defined through use practices, and their very existence relies on the continued practice of use, maintenance and self-organisation by a distributed group of users (Henfridsson and Bygstad 2013).

The organisation of digital infrastructures takes place through a distributed process in which multiple users, through application of skilled knowledge and ongoing negotiation, contribute to the underlying infrastructure (Yoo et al. 2012). The range of competences necessary for successful institutional change to occur far exceeds the capabilities of a single actor (Yoo et al. 2012). These distributed actors contribute to this infrastructure through (hard, soft and pseudo) code changes, as well as new adoption practices (enabled and constrained by said code).

These changes are often hotly contested at the level of adoption and use. Indeed, the Bitcoin community, like other OS communities, discourages rogue changes through political ideology and community backlash (Dahlander and Magnusson 2005; Kirsch 1997). The OS nature of the code, however, means that on a technical level forking cannot be prevented—and thus the new patterns of organizing that result cannot be prevented either.

Previous research has emphasised the embeddedness of digital infrastructures in the organisations which initially design and later rely on them (Hanseth and Lyytinen 2010; Henfridsson and Bygstad 2013; Star and Ruhleder 1996). This extant research sees organisational change as leading to infrastructural evolution, rather than vice versa. By showing how self-organisation takes place through code forking in online communities without pre-existing organisational structure, we aim to contribute a better understanding of the role of the digital in self-organising in response to previous calls for research on the role of digital materiality (Leonardi 2010).

While previous studies of infrastructural evolution have shown how infrastructure changes as a result of organisational changes, this empirical work shows the reverse: how the infrastructure—whether through change or existing composition—at the level of code leads to new organising practices. While user interpretations and adoption of the code is a vital part of this process, the possibilities open to the user are defined according to the code-level composition of the infrastructure. Fundamental changes, or what we have called speciation, therefore require changes to the fabric of the infrastructure through hard forks. The purpose of this study was to understand how this occurred.

Conclusion and Future Research

This paper provides an examination of the role of code in digital infrastructures that begins to build an understanding of how the act of coding digital infrastructure can lead to new patterns of self-organising.

This paper therefore contributes to extant research on digital infrastructure in the following ways. First, it identifies and conceptualises a new mode of organising around digital infrastructures where code, and forking, is a necessary pre-condition for changes in organisational structures, practices and composition. Second, based on literature on code development in OS communities, it proposes, substantiates and empirically identifies the concept of forking to show how, and when, code development practices combine into an organisational change mechanism. In so doing, it describes a vocabulary for describing the different patterns of self-organisation in online communities related to digital infrastructure in conceptualising forking as variation, adaptation and speciation.

These patterns of self-organising are most prominent among OS communities because members of the organisation themselves hold the skills, and tools, to enable code forks—a mechanism for organising in a digital-first world. While code-level, and thus organisational changes, are often hotly contested, both this paper and previous research in OS communities show that forking is sometimes unavoidable. As coding becomes more accessible and ubiquitous to non-engineering tasks, for instance through improved user interfaces and automation and more accessible programming languages, it is likely that the patterns of self-organising will become more common, and thus more deserving of close study. We therefore encourage future examination of the role of code, and code forking in particular, in organisational change and self-organising.

References

Aanestad, M., and Jensen, T. B. 2011. "Building nation-wide information infrastructures in healthcare through modular implementation strategies," *Journal of Strategic Information Systems* (20:2), Elsevier B.V., pp. 161–176 (doi: 10.1016/j.jsis.2011.03.006).

Aho, A. V, and Hopcroft, J. E. 1974. *Design & Analysis of Computer Algorithms*, Pearson Education India.

Andersen, J. V., Lindberg, A., Lindgren, R., and Selander, L. 2016. "Algorithmic Agency in Information Systems: Research Opportunities for Data Analytics of Digital Traces," in *Proceedings of the 49th Annual Hawaii International Conference on System Sciences*.

Ashcraft, K. L., Kuhn, T. R., and Cooren, F. 2009. "Constitutional Amendments: 'Materializing' Organizational Communication," *The Academy of Management Annals* (3:1), pp. 1–64 (doi: 10.1080/19416520903047186).

Austin, R. D., Devin, L., and Sullivan, E. E. 2011. "Accidental Innovation: Supporting Valuable Unpredictability in the Creative Process," *Organization Science* (23:5), pp. 1505–1522 (doi: 10.1287/orsc.1110.0681).

Barrett, M., Oborn, E., and Orlikowski, W. 2016. "Creating value in online communities: the sociomaterial configuring of strategy, platform, and stakeholder engagement," *Information Systems Research* (27:4), pp. 704–723.

Bechky, B. A. 2006. "Talking about machines, thick description, and knowledge work," *Organization Studies* (27:12), Sage Publications, pp. 1757–1768.

Beckert, J. 1999. "Agency, Entrepreneurs, and Institutional Change. The Role of Strategic Choice and

Institutionalized Practices in Organizations," *Organization Studies* (20:5), pp. 777–799.

Blei, D. M., Ng, A. Y., and Jordan, M. I. 2003. "Latent dirichlet allocation," *The Journal of Machine Learning Research* (3:1), pp. 993–1022 (doi: 10.1162/jmlr.2003.3.4-5.993).

Braa, J., Hanseth, O., Heywood, A., Woinshet, M., and Shaw, V. 2007. "Developing Health Information Systems in Developing Countries : the flexible Standards Strategy," *Management Information Systems Quarterly* (31:Suppl 1), pp. 381–402 (doi: 10.2307/25148796).

Broadbent, M., and Weill, P. 1997. "Management by maxim: How business and IT managers can create IT infrastructures," *Sloan Management Review* (38:3), pp. 77–92 (doi: Article).

Callon, M. 1986. "Some elements of a sociology of translation: domestication of the scallops and the fishermen of St Brieuc Bay," in *Power, action and belief: A new sociology of knowledge?*J. Law (ed.) (Vol. 32), Routledge, pp. 196–233.

Child, J. 1997. "Strategic Choice in the Analysis of Action, Structure, Organizations and Environment: Retrospect and Prospect," *Organization Studies* (18:1), pp. 43–76 (doi: 10.1177/017084069701800104).

Chuang, J., Manning, C. D., and Heer, J. 2012. "Termite : Visualization Techniques for Assessing Textual Topic Models Categories and Subject Descriptors," *International Conference on Advanced Visual Interfaces (AVI)* (doi: 10.1145/2254556.2254572).

Ciborra, C. 2000. *From control to drift: the dynamics of corporate information infastructures*, Oxford University Press on Demand.

Dahlander, L., and Magnusson, M. G. 2005. "Relationships between open source software companies and communities: Observations from Nordic firms," *Research Policy* (34:4), pp. 481–493 (doi: 10.1016/j.respol.2005.02.003).

Dhanarag, C., and Parkhe, A. 2006. "Orchestrating Innovation Networks," *Academy of Management Review* (31:3), pp. 659–669.

Eaton, B., Elaluf-Calderwood, S., Sørensen, C., and Yoo, Y. 2015. "Distributed Tuning of Boundary Resources: the Case of Apple's Ios Service System," *MIS Quarterly* (39:1), pp. 217–243.

Edwards, P. N., Jackson, S. J., Bowker, G. C., and Knobel, C. 2007. "Understanding infrastructure: Dynamics, tensions, and design.," *Report of a Workshop on "History & Theory of Infrastructure: Lessons for New Scientific Cyberinfrastructures"* (doi: 2027.42/49353).

Fang, Y., and Neufeld, D. 2009. "Understanding Sustained Participation in Open Source Software Projects," *Journal of Management Information Systems* (25:4), pp. 9–50 (doi: 10.2753/MIS0742-1222250401).

Gaskin, J., Berente, N., Lyytinen, K., and Yoo, Y. 2014. "Toward Generalizable Sociomaterial Inquiry: a Computational Approach for Zooming in and Out of Sociomaterial Routines.," *MIS Quarterly* (38:3).

Gherardi, S. 2009. "Introduction: The Critical Power of the `Practice Lens'," *Management Learning* (40:2), pp. 115–128 (doi: 10.1177/1350507608101225).

Glaser, B. G., and Strauss, A. L. 1999. *The Discovery of Grounded Theory: Strategies for qualitative research*, New Brunswick: AldineTransaction.

Hanseth, O., and Aanestad, M. 2003. "Design as bootstrapping. On the evolution of ICT networks in health care," *Methods of information in medicine* (42:4), FK SCHATTAUER VERLAGSGESELLSCHAFT MBH, pp. 384–391.

Hanseth, O., Jacucci, E., Grisot, M., and Aanestad, M. 2006. "Reflexive Standardization: Side Effects and Complexity in Standard Making," (30), pp. 563–581.

Hanseth, O., and Lyytinen, K. 2010. "Design theory for dynamic complexity in information infrastructures: the case of building internet," *Journal of Information Technology* (25:1), Palgrave Macmillan, pp. 1–19 (doi: 10.1057/jit.2009.19).

Hanseth, O., and Monteiro, E. 1997. "Inscribing behaviour in information infrastructure standards," *Accounting, Management and Information Technologies* (7:4), pp. 183–211.

Hearn, M. 2015. "Why is Bitcoin forking? A tale of differing visions," (available at https://medium.com/faith-and-future/why-is-bitcoin-forking-d647312d22c1#.8dq5fi3we).

Hedman, J., Srinivisan, N., and Lindgren, R. 2013. "Digital Traces of Information Systems: Sociomateriality Made Researchable," in *Thirty Fourth International Conference on Information Systems, Milan 2013* (Vol. 38), pp. 809–830.

Henfridsson, O., and Bygstad, B. 2013. "The generative mechanisms of digital infrastructure evolution," *MIS Quarterly* (37:3), pp. 907–931.

Holland, J. H. 1995. *Hidden order: How adaptation builds complexity*, Basic Books.

Howison, J., Wiggins, A., and Crowston, K. 2011. "Validity Issues in the Use of Social Network Analysis with Digital Trace Data," *Journal of the Association for Information Systems* (12:12), pp. 767–797.

Iannacci, F. 2010. "When is an information infrastructure? Investigating the emergence of public sector information infrastructures," *European Journal of Information Systems* (19:1), pp. 35–48 (doi: 10.1057/ejis.2010.3).

Kartseva, V., Hulstijn, J., Gordijn, J., and Tan, Y.-H. 2010. "Control patterns in a health-care network," *European Journal of Information Systems* (19:3), pp. 320–343 (doi: 10.1057/ejis.2010.13).

Kirsch, L. S. 1997. "Portfolios of control modes and IS project management," *Information Systems Research* (8:3), pp. 215–239.

Kitagaki, I., and Hikita, A. 2007. "Development of an algorithm for groupware modeling for a collaborative learning," *International Journal of Computers, Communications & Control* (1).

Kuk, G., and Janssen, M. 2013. "Assembling infrastructures and business models for service design and innovation," *Information Systems Journal* (23:5), pp. 445–469 (doi: 10.1111/j.1365-2575.2012.00418.x).

Latour, B. 1987. *Science in action: How to follow scientists and engineers through society*, Harvard university press.

Lave, J., and Wenger, E. 1991. *Situated learning: Legitimate peripheral participation*, BOOK, Cambridge university press.

Leonardi, P. M. 2010. "Digital materiality? How artifacts without matter, matter," *First Monday* (15:6).

Leonardi, P. M. 2011. "When Flexible Routines Meet Flexible Technologies: Affordance, Constraint, and the Imbrication of Human and Material Agencies.," *MIS Quarterly* (35:1), pp. 147–168 (doi: 1005).

Leonardi, P. M. 2013. "The Emergence of Materiality within Formal Organizations," in *How Matter Matters: Objects, Atrifacts and Materiality in Organisation Studies*P. R. Carlile, D. Nicolini, A. Langley, and H. Tsoukas (eds.), Oxford: Oxford University Press, pp. 142–170.

Malhotra, N. K., Kim, S. S., and Agarwal, J. 2004. "Internet users' information privacy concerns (IUIPC): The construct, the scale, and a causal model," *Information Systems Research* (15:4), pp. 336–355 (doi: 10.1287/isre.1040.0032).

Meeker, H. J. 2008. *The open source alternative: understanding risks and leveraging opportunities*, John Wiley & Sons.

Miles, M. B., and Huberman, A. M. 1994. *Qualitative data analysis: An expanded sourcebook*, Sage.

Mol, A., and Law, J. 2002. "Complexities: an introduction," Duke University Press.

Morisse, M., and Ingram, C. 2016. "A mixed blessing: Resilience in the entrepreneurial socio-technical sysetm of Bitcoin," *Journal of Information Systems and Technology Management* (13:1).

Nakamoto, S. 2008. "Bitcoin: A peer-to-peer electronic cash system," (doi: 10.1007/s10838-008-9062-0).

Nan, N. 2011. "Capturing Bottom-up Information Technology Use Processes: A Complex Adaptive

Systems Model," *MIS Quarterly* (35:2), pp. 505–532.

Nyman, L. 2015. "Understanding Code Forking in Open Source Software," Hanken School of Economics.

Nyman, L., and Lindman, J. 2013. "Code Forking, Governance, and Sustainability in Open Source Software," *Technology Information Management* (January), pp. 7–12.

Orlikowski, W. J. 2007. "Sociomaterial Practices: Exploring Technology at Work," *Organization Studies* (28:9), pp. 1435–1448 (doi: 10.1177/0170840607081138).

Orlikowski, W. J., and Robey, D. 1991. "Information Technology and the Structuring of Organizations," *Information Systems Research* (2:2), pp. 143–169.

Phang, C. W., Kankanhalli, A., and Huang, L. 2014. "Drivers of Quantity and Quality of Participation in Online Policy Deliberation Forums," *Journal of Management Information Systems* (31:3), pp. 172–212 (doi: 10.1080/07421222.2014.995549).

Phang, C. W., Kankanhalli, A., and Tan, B. C. Y. 2015. "What Motivates Contributors vs. Lurkers? An Investigation of Online Feedback Forums," *Information Systems Research* (26:4), pp. 773–792 (doi: 10.1287/isre.2015.0599).

Pipek, V., and Wulf, V. 2009. "Infrastructuring: Toward an Integrated Perspective on the Design and Use of Information Technology," *Journal of the Association for Information Systems* (10:May 2009), pp. 447–473 (doi: Article).

Rai, A., Patnayakuni, R., and Seth, N. 2006. "Firm performance impacts of digitally enabled supply chain integration capabilities," *MIS quarterly* (30:2), pp. 225–246.

Raymond, E. 1999. "The cathedral and the bazaar," *Knowledge, Technology & Policy* (12:3), Springer, pp. 23–49.

Ren, Y., Harper, F. M., Drenner, S., Terveen, L., Kiesler, S., Riedl, J., and Kraut, R. E. 2012. "Building member attachment in online communities: Applying theories of group identity and interpersonal bonds," *Mis Quarterly* (36:3), pp. 841–864.

Robles, G., and González-Barahona, J. M. 2012. "A comprehensive study of software forks: Dates, reasons and outcomes," in *IFIP International Conference on Open Source Systems*, Springer, pp. 1–14.

Sievert, C., and Shirley, K. 2014. "LDAvis: A method for visualizing and interpreting topics," in *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pp. 63–70.

Siggelkow, N. 2007. "Persuasion with case studies," *Academy of Management Journal* (50:1), pp. 20–24 (doi: 10.5465/AMJ.2007.24160882).

Star, S. L. 1999. "The Ethnography of Infrastructure," *American Behavioral Scientist* (43:3), pp. 377–391 (doi: 10.1177/00027649921955326).

Star, S. L., and Ruhleder, K. 1996. "Steps toward an ecology of infrastructure: Design and access for large information spaces," *Information systems research* (7:1), pp. 111–134.

Tanriverdi, H., Konana, P., and Ge, L. 2007. "The choice of sourcing mechanisms for business processes," *Information Systems Research* (18:3), pp. 280–299.

Taylor, M. B. 2013. "Bitcoin and The Age of Bespoke Silicon How Bitcoin Works : User Perspective Bitcoin Mining : Miner's Perspective," in *Proceedings of 2013 International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES)*, Montreal, Canada, pp. 1–10.

Tilson, D., Lyytinen, K., and Sørensen, C. 2010. "Digital infrastructures: The missing IS research agenda," *Information Systems Research* (21:4), pp. 748–759 (doi: 10.1287/isre.1100.0318).

Vaast, E., and Walsham, G. 2009. "Trans-situated learning: supporting a network of practice with an information infrastructure," *Information Systems Research* (20:4), pp. 547–564.

Venkatesh, V., Brown, S. a., and Bala, H. 2013. "Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in Information Systems," *Management Information Systems Quarterly* (37:3), pp. 855–879.

Weber, S. 2004. *The success of open source* (Vol. 368), Cambridge Univ Press.

Wenger, E. 2000. "Communities of Practice and Social Learning Systems," *Organization* (7:2), pp. 225–246 (doi: 0803973233).

Yin, R. 2003. *Applications of Case Study Research (Applied Social Research Methods)* (4th editio.), Thousand Oaks: Sage .

Yoo, Y., Boland, R., Lyytinen, K., and Majchrzak, A. 2012. "Organizing for Innovation in the Digitized World," *Organization Science* (23:5), pp. 13–32.

Yoo, Y., Henfridsson, O., and Lyytinen, K. 2010. "The new organizing logic of digital innovation: An agenda for information systems research," *Information Systems Research* (21:4), pp. 724–735 (doi: 10.1287/isre.1100.0322).

Yoo, Y., Lyytinen, K., and Boland Jr., R. J. 2008. "Distributed Innovation in Classes of Networks," in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, Ieee, January, pp. 58–58 (doi: 10.1109/HICSS.2008.125).

Yoo, Y., Lyytinen, K., and Yang, H. 2005. "The role of standards in innovation and diffusion of broadband mobile services: The case of South Korea," *Journal of Strategic Information Systems* (14:3), pp. 323–353 (doi: 10.1016/j.jsis.2005.07.007).

Zimmer, C., and Emlen, D. J. 2013. *Evolution: Making sense of life*, Roberts Greenwood Village, CO.

Zittrain, J. L. 2006. "The Generative Internet," *Harvard Law Review* (119), pp. 1974–2040.