

Copyright 2014 by Shawn Bayern

Printed in U.S.A.
Vol. 108, No. 4

Online Essay

OF BITCOINS, INDEPENDENTLY WEALTHY SOFTWARE, AND THE ZERO-MEMBER LLC[†]

Shawn Bayern

ABSTRACT—An innovative software technology known as Bitcoin makes it easier for software to operate with some degree of financial autonomy. In a meaningful sense, it is now possible for software to conduct business on its own account, without using the traditional financial system as an intermediary and without a financial existence tied to an existing natural or legal person. This Essay explores this possibility and suggests that legally autonomous entities, such as a limited liability company (LLC) with no members, are a useful legal structure for factually autonomous systems.

AUTHOR—Larry & Joyce Beltz Professor, Florida State University College of Law; B.S., Yale University, Computer Science; J.D., University of California, Berkeley School of Law. Thanks to Rob Atkinson, David Landau, David Russcol, Don Weidner, and Sam Wiseman for helpful discussions.

[†] This Essay was originally published in the Northwestern University Law Review Online on April 10, 2014, 108 NW. U. L. REV. ONLINE 257 (2014), <http://www.law.northwestern.edu/depts/legalpub/lawreview/online/2014/6/Bayern.pdf>.

INTRODUCTION	1486
I. BITCOIN: THE IMPORTANT DETAILS	1487
II. INDEPENDENTLY WEALTHY SOFTWARE	1492
III. THE LAW OF AUTONOMOUS ORGANIZATIONS.....	1495
A. <i>Autonomous Legal Entities</i>	1495
B. <i>Common Law and Regulatory Challenges</i>	1498

INTRODUCTION

An innovative software technology known as Bitcoin¹ has been in the news recently, though mostly for the wrong reasons. The financial press, as it typically does, aims to fuel speculative predictions on future market prices, and Bitcoin has captured the press's attention largely because of an extremely rapid growth in the price of the scarce units of account that the Bitcoin technology creates. Worth essentially nothing four years ago, bitcoins recently have helped individuals make and lose fortunes.²

This Essay is not concerned with the price of a bitcoin; indeed, though I have followed the technical details of the software from its early days, I have no useful personal price speculation to offer, and I doubt anyone else does either. In fact, it is surprising how readily experts continue to try to predict the financial future.³

Instead, this Essay introduces to the legal community a fascinating, decreasingly farfetched technological possibility that the Bitcoin software promotes, and it offers suggestions for how the law might interact with that possibility. In short, Bitcoin allows autonomously operating software—such as a computer virus or the software that manages a network of vending machines—to exercise control over significant wealth, not as an intermediary for individuals or companies, but rather, in a functionally meaningful sense, in its own right.

¹ See *Bitcoin*, BITCOIN, <https://bitcoin.org/en/> (last visited Mar. 30, 2014) (“Bitcoin uses peer-to-peer technology to operate with no central authority or banks; managing transactions and the issuing of bitcoins is carried out collectively by the network.”).

² See, e.g., Nathaniel Popper, *Investors, and Swindlers, See Big Profits in Bitcoin*, INT’L N.Y. TIMES, Dec. 7, 2013, at 14.

³ See, e.g., Eric Posner, *Fool’s Gold: Bitcoin is a Ponzi Scheme—The Internet’s Favorite Currency Will Collapse*, SLATE (Apr. 11, 2013, 11:11 AM), http://www.slate.com/articles/news_and_politics/view_from_chicago/2013/04/bitcoin_is_a_ponzi_scheme_the_internet_currency_will_collapse.html (“Unless a bitcoin has value as a currency, it has no value at all, and its price in dollars will fall to zero. . . . Bitcoin will collapse when people realize that it can’t survive as a currency because of its built-in deflationary features, or because of the emergence of [an alternative], or both.”). For perspective, I continue to find Grant Gilmore’s skepticism of strong conclusions from the social sciences helpful: “[N]o historian, social scientist or legal theorist has ever succeeded in predicting anything.” GRANT GILMORE, *THE AGES OF AMERICAN LAW* 99 (1977).

Part I of this Essay briefly describes the Bitcoin software technology for a legal audience. Part II explains how Bitcoin promotes the possibility of independently wealthy software. Part III considers several legal approaches to regulating autonomous software that acts as a business enterprise. The legal analysis is exploratory, and it is only a first step; it is too early to offer definite normative opinions. But it will likely be important for legal “technologies,” such as organizational forms; concepts of legal entities and legal personality; and institutional systems of contract and tort law, to keep pace with technological innovation.⁴

I. BITCOIN: THE IMPORTANT DETAILS

The Bitcoin software is an attempt to provide for the distributed management, by computer software, of artificially scarce resources known as bitcoins. If the system works, as it has so far been shown to do, it permits people to send these scarce digital resources at low cost to one another over the public Internet without relying on a trusted intermediary. The resources themselves have negligible “intrinsic” value, much like conventional modern national currencies and even (putting aside minor industrial uses) gold. But market forces have, for some time now, treated them as valuable. For our purposes, the precise value that the market assigns to the scarce resources controlled by Bitcoin is not important; all that is important is that “a bitcoin”—an arbitrary, divisible unit of the scarce resources created by the software—has some value.

So long as (1) the system continues to work as expected, and (2) market participants continue to assign a positive value to a bitcoin, anyone with an Internet-connected computer can, because of the Bitcoin software, send something of financial value to others, anywhere in the world, without relying on an intermediary institution. This is true even if that person has no wealth outside the Bitcoin system, no access to local bank accounts or other financial organizations and, for many practical purposes, no permission from local regulatory authorities. The ability to transfer financial value in this way, in short, is the software’s significant technological innovation.

⁴ Bitcoin raises other problems for legal regulation that need to be addressed in their own right. There are a variety of important but comparatively mundane questions, such as what sort of capital gains treatment, if any, should attend profits from private trading in bitcoins. More subtle is the potential tax treatment of profits from “mining,” or creating through software processes the valuable units of account in cryptographically backed currency. There are also questions about the interaction between Bitcoin and the securities laws. *See, e.g.*, SEC v. Shavers, No. 4:13-CV-416, 2013 WL 4028182 (E.D. Tex. Aug. 6, 2013). And systems like Bitcoin surely will raise both challenges and possibilities for criminal law enforcement. *See, e.g.*, Press Release, U.S. Attorney’s Office, S. Dist. of N.Y., Manhattan U.S. Attorney Announces Charges Against Bitcoin Exchangers, Including CEO of Bitcoin Exchange Company, for Scheme to Sell and Launder over \$1 Million in Bitcoins Related to Silk Road Drug Trafficking (Jan. 27, 2014) (*available at* <http://www.justice.gov/usao/nys/pressreleases/January14/SchremFaiellaChargesPR.php>).

Bitcoin was developed between 2007 and 2009 by a programmer called Satoshi Nakamoto, though there is consensus that that was a pseudonym.⁵ Nakamoto initially informed an established group of cryptographers of his idea⁶ and then released an implementation of it to that group,⁷ eventually setting up a forum for discussing and promoting Bitcoin.⁸ Nakamoto, however, no longer actively communicates or participates in Bitcoin under that name, and he remains unidentified. Instead, Bitcoin's development has proceeded in a typical fashion for an open-source project, with a variety of volunteer programmers⁹ maintaining the software under informal processes that depend on rough notions of consensus and that are subject to no fixed legal or organizational structure.¹⁰ People download and run new versions of the software largely because they trust its ongoing process of development. As with many open-source, networked software projects, there can be—and are—multiple versions of the software, written by distinct parties, that can all interoperate.¹¹

Bitcoin is a *peer-to-peer software system*, which means, practically speaking, that the entire system is made up of versions of the software that end-users download and run on their personal computers. There is no Bitcoin server or Bitcoin company that directly manages the system. Indeed, the lack of such centripetal features was a core design goal for Bitcoin; as Nakamoto once wrote, “At some point I became convinced

⁵ See Reuben Grinberg, *Bitcoin: An Innovative Alternative Digital Currency*, 4 HASTINGS SCI. & TECH. L.J. 159, 162 (2012); Posting of Satoshi Nakamoto to cryptography@metzdowd.com (Nov. 1, 2008), <http://www.mail-archive.com/cryptography@metzdowd.com/msg09959.html>.

⁶ See Posting of Satoshi Nakamoto, *supra* note 5.

⁷ See Posting of Satoshi Nakamoto to cryptography@metzdowd.com (Jan. 9, 2009), <http://www.mail-archive.com/cryptography@metzdowd.com/msg10142.html>.

⁸ See BITCOIN FORUM, <https://bitcointalk.org/> (last visited Mar. 30, 2014).

⁹ The current lead developer of Bitcoin is now apparently compensated by a group calling itself the Bitcoin Foundation, which accepts donations in bitcoins and claims to pay some of them to the developer. The legal structure of that organization is unclear to the public. For more information, see *Governance Structure*, BITCOIN FOUNDATION, <https://bitcoinfoundation.org/about/governance> (last visited Mar. 30, 2014).

¹⁰ For a somewhat technical study of the structure of modern open-source software development, see Christian Bird et al., *Latent Social Structure in Open Source Projects*, in PROCEEDINGS OF THE 16TH ACM SIGSOFT INTERNATIONAL SYMPOSIUM ON FOUNDATIONS OF SOFTWARE ENGINEERING (Mary Jean Harrold & Gail C. Murphy eds., 2008), available at <http://macbeth.cs.ucdavis.edu/cathbaz.pdf>.

¹¹ Nakamoto, however, was initially opposed to multiple, interoperable implementations of Bitcoin. See Satoshi Nakamoto, *Re: Transactions and Scripts: DUP HASH160... EQUALVERIFY CHECKSIG*, BITCOIN FORUM (June 17, 2010, 6:46 PM), <https://bitcointalk.org/index.php?topic=195.msg1611#msg1611> (“I don’t believe a second, compatible implementation of Bitcoin will ever be a good idea. So much of the design depends on all nodes getting exactly identical results in lockstep that a second implementation would be a menace to the network.”).

there was a way to do this without any trust required at all and couldn't resist to keep thinking about it."¹²

Most of Bitcoin's features stem from its decision to avoid what network theorists call a *trusted third party*—that is, an authority, above reproach, that can inform others of the canonical state of the system. Consider how easy it would be, relatively speaking, to design an Internet-based currency if the design permitted a party that everyone trusts to coordinate its operation: the trusted party would issue the digital money according to generally accepted criteria, verify its authenticity, manage its exchange, and so on. Amazon gift cards work in roughly this way; people can purchase them from Amazon, log in to check their balance, and apply them toward future orders. Such a system is extremely straightforward to create. Though its technical design would of course need to be appropriately secure, it is neither an academic nor a practical challenge to create it; it is inexpensive and simple. For example, if the U.S. government, or a consortium of major American banks, wished to make it easier, faster, or cheaper than it is today to move dollars around the world online, they could do so without significant technological innovation.

Bitcoin is different. It is unclear precisely what motivated Nakamoto to design a decentralized financial system.¹³ Perhaps it was an academic challenge. Partly, however, Nakamoto seems to have intended to make it robust against certain kinds of organized disruptions: “Governments are good at cutting off the heads of a centrally controlled network[] like Napster, but pure [peer-to-peer] networks like Gnutella and Tor seem to be holding their own.”¹⁴ Whatever the case, unlike a centrally managed payment system like PayPal, there is a meaningful sense in which nobody is in charge of Bitcoin.

¹² Satoshi Nakamoto, *Re: Transactions and Scripts: DUP HASH160... EQUALVERIFY CHECKSIG*, BITCOIN FORUM (June 18, 2010, 4:17 PM), <https://bitcointalk.org/index.php?topic=195.msg1617#msg1617>.

¹³ A prevailing opinion in some of the Internet forums in which people commonly discuss Bitcoin is that Nakamoto was motivated by libertarian political opinions, given Bitcoin's ability to avoid some political barriers to trade. Nakamoto's writings acknowledge the attractiveness of Bitcoin to libertarians, but they do not specifically support any view of his own politics. See Posting of Satoshi Nakamoto to cryptography@metzdowd.com (Nov. 14, 2008), <http://www.mail-archive.com/cryptography@metzdowd.com/msg10001.html> (“[The bitcoin system is] very attractive to the libertarian viewpoint if we can explain it properly.”).

¹⁴ Posting of Satoshi Nakamoto to cryptography@metzdowd.com (Nov. 7, 2008), <http://www.mail-archive.com/cryptography@metzdowd.com/msg09971.html>. At the time Nakamoto wrote this, Gnutella was probably the most popular file-sharing network, commonly used to share unlicensed, copyrighted material. See Eric Bangeman, *Study: BitTorrent Sees Big Growth, LimeWire Still #1 P2P App*, ARS TECHNICA, (Apr. 21, 2008, 7:32 PM), <http://arstechnica.com/uncategorized/2008/04/study-bittorrent-sees-big-growth-limewire-still-1-p2p-app/>. Tor is a network originally developed by the U.S. Naval Research Laboratory that permits plausibly deniable anonymity over the public Internet. See *Tor: Overview*, TOR, <https://www.torproject.org/about/overview.html.en> (last visited Mar. 30, 2014).

It may seem counterintuitive that a decentralized online network could ever be an appropriate mechanism for storing and updating convergent financial records. After all, what prevents cheating? The most obvious kind of cheating might seem to be counterfeiting, but this problem is relatively easy to solve through modern cryptography. Conventional uses of cryptography, such as those that encrypt the network connection between a financial institution and its individual clients, can ensure the authenticity of a communication. Just as someone without my password could not log into my credit union account, someone without my private Bitcoin keys could not spend my bitcoins.¹⁵

A decentralized online financial network raises a more difficult problem than ensuring authenticity: What prevents a person from trying to transmit the same digital asset twice to two different people? Suppose I enter a physical storefront and convince the storeowner that I have \$100 worth of online currency and wish to transfer it to him. Without a central arbiter to confirm that I have committed a transfer of funds to the owner's account, merely giving him something like "access to the funds" has a potentially complicated meaning. What if, at around the same time I give him the digital key that confers access to my funds, I send an email with the same key to an organization across the world, telling it that I wish to spend \$100 at *its* store? Without an authoritative arbiter, how would one recipient know that he was the legitimate new owner of my funds, and how would the other know that it was the victim of an attempt to spend the money twice?

Much of Bitcoin's complexity addresses this problem. For years, it was thought that an online financial network would require at least some central locus of trusted authority in order to prevent duplicate transactions.¹⁶ Nakamoto's software, however, showed that such trust was unnecessary; decentralized software could reliably agree upon a single, authoritative sequence of records so that each potential recipient of funds could know that he or she is the only recipient of those funds.

The precise mechanism by which Bitcoin produces this authoritative sequence is complex, but in short, and loosely speaking, it allows participants to add new financial records to the authoritative sequence by demonstrating that they have expended computing power on an otherwise unimportant, repetitive task. This process, known as Bitcoin mining, confers the right to add a record to the sequence (and also, not incidentally, it is rewarded by the creation of new bitcoins, partly as an incentive to participate in the network and partly as a way to manage the initial

¹⁵ See *Some Bitcoin Words You Might Hear*, BITCOIN, <https://bitcoin.org/en/vocabulary> (last visited Mar. 30, 2014) ("In the case of Bitcoin, cryptography is used to make it impossible for anybody to spend funds from another user's wallet or to corrupt the block chain.").

¹⁶ See SATOSHI NAKAMOTO, BITCOIN: A PEER-TO-PEER ELECTRONIC CASH SYSTEM 2 (2008), available at <http://bitcoin.org/bitcoin.pdf>.

distribution of bitcoins).¹⁷ In the event of a dispute among different candidate sequences of transactions, the one that is eventually backed by the most computing power wins.

Accordingly, for parts of its operation, Bitcoin trusts the majority of the computing power of the network on a principle that could roughly, and a bit imprecisely, be called “one computer-processor operation, one vote.” Thus, faster computers have more influence than slower ones, specialized hardware can dwarf the “voting” power of ordinary computers, and so on. As time goes on and more powerful devices run legitimate copies of the software, it becomes extremely difficult for any single party to disrupt the system. As a result, widely distributed parties can agree on a canonical financial ledger even though no one is “in charge” of it.

In practice, this description of Bitcoin is not quite accurate because Bitcoin does not operate in as rigorously decentralized a manner as Nakamoto originally designed it.¹⁸ For example, when people download a copy of the Bitcoin software that participates in the overall network, they also commonly download data, produced by the current developers of the peer-to-peer software, that serves to confirm the version of the history of the network that the current developers deem to be authoritative. Without this *checkpointing*, fraud against the Bitcoin network would be at least conceptually possible if someone were to amass enough computing power to dwarf the computing power already expended by participants in the Bitcoin network over the last several years. This would be a significant practical challenge given the present and historical size of the Bitcoin network, but it would perhaps not be impossible for a well-funded attacker with specialized hardware.

The Bitcoin network has moved away from an entirely decentralized model for other reasons. The developers of the Bitcoin client have the ongoing capacity to change the Bitcoin protocol in minor but incompatible ways, actively managing the community of Bitcoin users to make sure that the Bitcoin network upgrades in ways they have determined. Bitcoin arguably has already changed more than Nakamoto expected,¹⁹ but the principles behind its design have remained the same.

In short, the Bitcoin network has operated for some time as a reliable and fraud-resistant means for transmitting value online without financial intermediaries. Even more importantly, for the purposes of this Essay, it

¹⁷ See *id.* at 4.

¹⁸ Cf. BEN LAURIE, DECENTRALISED CURRENCIES ARE PROBABLY IMPOSSIBLE: BUT LET’S AT LEAST MAKE THEM EFFICIENT 3 (2011), available at <http://www.links.org/files/decentralised-currencies.pdf> (“If Bitcoin is, indeed, using a known consensus group, then it has, after all, a central authority (that consensus group), and is not, therefore, a decentralised currency.”).

¹⁹ See Nakamoto, *supra* note 11 (“The nature of Bitcoin is such that once version 0.1 was released, the core design was set in stone for the rest of its lifetime. Because of that, I wanted to design it to support every possible transaction type I could think of.”). Already, perhaps depending on what “core design” means, Bitcoin has been shown to be more flexible than this view suggests.

empowers software to transmit value without interacting with the existing financial system—and perhaps, as it turns out, without even interacting with humans.

II. INDEPENDENTLY WEALTHY SOFTWARE

The existence of computer programs that act autonomously, disconnected from direct human oversight, is no longer confined to fiction. Indeed, examples abound of software systems that could be described as autonomous. Most computer viruses that spread themselves over the Internet, propagating themselves by infecting other computers, qualify.²⁰ Google’s self-driving cars theoretically qualify, at least in a limited sense. Much software on and off the Internet can operate for years with minimal oversight.

There is, of course, already little to prevent a barely overseen software system from transferring significant amounts of money. Automated trading in financial markets is now commonplace. When automated trading functions properly, it accounts for a significant part of the operation of major financial institutions.²¹ When it does not, it causes phenomena that would have seemed astounding just ten years ago, like the “Flash Crash” of 2010 in which the American stock market briefly lost almost ten percent of its value.²² But because such software is ultimately acting rather directly on behalf of people and legal organizations, we do not think of it as owning its resources. It operates, for example, on financial accounts that are legally registered in the name of individuals or companies.

Bitcoin’s decentralized financial system raises an intriguing possibility. It does not depend directly on the law’s recognition of such things as bank accounts or the law’s requirements that financial institutions verify their customers.²³ These features of the system make way for software that could be programmed to act as if it were conducting business on its own account. It might, as it were, become independently wealthy.

Almost thirty years ago, Meir Dan-Cohen suggested the possibility of a self-owning company.²⁴ Conceiving of the modern corporation as “a

²⁰ This sort of virus is often called, in technical circles, a *worm*. See *Worm*, FREE ONLINE DICTIONARY OF COMPUTING, <http://foldoc.org/worm> (last updated Sept. 17, 1996) (defining “worm” as “[a] program that propagates itself over a network, reproducing itself as it goes”).

²¹ See generally Tom C.W. Lin, *The New Investors*, 60 UCLA L. REV. 678, 689–91 (2013) (describing the rise of algorithmic trading in modern finance).

²² See Tom Lauricella & Peter A. McKay, *Dow Takes a Harrowing 1010.14-Point Trip*, WALL ST. J. (May 7, 2010), <http://online.wsj.com/news/articles/SB10001424052748704370704575227754131412596>.

²³ E.g., 12 C.F.R. § 21.1–.21 (2013) (implementing statutory requirements to address money laundering).

²⁴ See MEIR DAN-COHEN, RIGHTS, PERSONS AND ORGANIZATIONS 46 (1986); see also Katsuhito Iwai, *Persons, Things and Corporations: The Corporate Personality Controversy and Comparative*

machine endowed with artificial intelligence,” he imagined a process whereby a corporation repurchases all its own outstanding shares, thereby becoming “ownerless.”²⁵ This can, of course, happen without Bitcoin; Dan-Cohen’s self-owning company did not rely on then-futuristic, decentralized currencies. And to be sure, the possibility of independently wealthy autonomous software could be achieved without the complex financial system provided by Bitcoin. For example, software could illegally use the identity of an individual to access a bank account. Or, even more simply, a legal and financial system managed by a trusted third party could choose not to require that bank accounts bear the title of an individual or legal entity; software could easily, for example, transact with a bank that identifies “owners” of accounts merely as those who demonstrate knowledge of a passcode.

But Bitcoin’s success as a decentralized, software-based financial instrument has paved the way for autonomous software to have convenient, legal access to a functionally independent financial life. It practically enables what in the past was just a theoretical possibility.

Consider an example. Suppose that an independently operating software program, which we might otherwise have classified as a computer virus, is designed to do something useful, like perform calculations or store backup copies of data for customers.²⁶ Without Bitcoin, there is no legal, convenient way for that software to accept payment from customers unless it does so on behalf of an individual or legal entity, such as a corporation, partnership, or trust. Because of Bitcoin, however, this program may send and receive valuable digital tokens from anyone on the Internet, and so it can easily act, in functional terms, as an independent business, accepting payment for its services and paying for the resources it uses (such as computer processing and storage) in order to provide them. It can make these choices as functions of its own software, without ongoing input from humans, and then execute the choices using online digital currencies like Bitcoin.

Importantly, an independently operating software program could likely do this—practically, very soon in the future, and indeed as a relatively minor engineering challenge—without special concessions from the law of any jurisdiction. So long as the inputs to its production are offered online in exchange for bitcoins, the program has no need for conventional financial institutions to pay its operational bills. So long as its customers are willing

Corporate Governance, 47 AM. J. COMP. L. 583, 599–600 (1999) (briefly describing the intellectual history of this idea and attributing it to others as well, including Martin Wolff and Alfred Conard).

²⁵ DAN-COHEN, *supra* note 24, at 49.

²⁶ Cf. Gmaxwell, *Re: Bitcoin the Enabler—Truly Autonomous Software Agents Roaming the Net*, BITCOIN FORUM (Dec. 7, 2011, 5:45 AM), <https://bitcointalk.org/index.php?topic=53855.msg642768#msg642768> (explaining concept of information storage platform utilizing Bitcoin as a payment currency).

to pay the software with bitcoins, it needn't use those institutions to manage its accounts receivable. Because of advances in computer-networking technology,²⁷ software can even hide its physical location relatively well, so it might even be difficult to interdict an autonomous system whose only interaction with the world is to produce and consume digital services (like storage or information processing) in exchange for bitcoins.

It may initially seem counterintuitive that anyone would set up such an independent business, given the opportunity to profit themselves from such businesses. But the Internet is a big place, and people's motivations are diverse. Some might do it simply for the technical challenges; others might contribute because they would prefer to work with an organization that aims to provide revenue-neutral pricing rather than to extract a profit. Depending on the service, manual oversight and the private profit motive may not be especially important; thus, for example, many local grocery stores are cooperatives, many hospitals and universities are not-for-profit entities, and many financial institutions are mutually owned.²⁸ It isn't hard to imagine an open-source developer writing the software needed to "seed" a self-operating online business or potential customers chipping in with initial funding because of the marketplace advantages they anticipate that it will bring them. People have paid more for less on "crowd-funding" sites like Kickstarter.²⁹

Intriguingly, once a system that provides online services is possible, there is little conceptual barrier to a similar system's providing more conventional "offline" services beyond the Internet. So, for example, consider a network of vending machines. Someone needs to install the vending machines and continuously supply them. But from the perspective of the software operating the network, those tasks are simply another type of input to production, like disk space or network bandwidth.³⁰ The software can pay someone to install or stock a new vending machine, verify that the task has been completed, and remit payment digitally using Bitcoin. Nothing prevents such services from interacting with each other and assuming a significant economic role.

Once an autonomous system begins to have a real-world presence, its legal regulation becomes potentially weightier. In a still fanciful future, for example, the autonomous software that operates a fleet of airborne drones

²⁷ See Lin, *supra* note 21.

²⁸ Cf. Shawn Bayern, *Toward Nonprofit Financial Services*, HUFFINGTON POST (May 12, 2009, 12:46 PM), http://www.huffingtonpost.com/shawn-bayern/toward-nonprofit-financia_b_201678.html (describing advantages of not-for-profit financial services such as Vanguard, TIAA-CREF, and credit unions).

²⁹ Indeed, Bitcoin can itself enable Kickstarter-like crowd-funding without trusted third parties. See Mike Hearn, Bitcoin Developer, *The Future of Money* (Aug. 23, 2013), available at <https://www.youtube.com/watch?v=Pu4PAMFPo5Y> (at 19:25).

³⁰ Cf. Gmaxwell, *supra* note 26.

that deliver packages³¹ might injure someone when one of its delivery devices accidentally collides with someone's car. I address this sort of problem in the next section.

III. THE LAW OF AUTONOMOUS ORGANIZATIONS

This section explores some of the legal ramifications of the potential rise of independently operating copies of business software. First, it considers their interaction with existing legal organizational forms, such as the limited liability company (LLC); in this area, the law, perhaps surprisingly, already offers appropriate legal "technologies" to respond to some of Bitcoin's possibilities. Second, it considers the role of contract and tort law in enabling and regulating autonomous systems.

A. *Autonomous Legal Entities*

The law, of course, has a rich history of recognizing fictional entities and endowing them with many of the same abilities as those of individuals. This technique, often taken for granted, has proven to be an extremely productive simplification of the interaction between organizational law and other areas of law, such as property law, contract law, and procedural law. We could imagine a system in which partnerships or LLCs need a distinctive mechanism to initiate or defend lawsuits, own property, or become parties to contracts;³² instead, we integrate organizational entities into the rest of the legal system by means of a convenient abstraction: we treat them, for many purposes, as if they were individuals.

This technique can be readily extended to accommodate businesses managed by autonomous software. Indeed, it has effectively already been extended in this way, though perhaps unintentionally.

First, however, there is a threshold question: Why would we want the legal system to accommodate autonomous software? Isn't there something unsettling about a soulless entity pursuing its own purposes and answerable to no one? Isn't it problematic for a computer program to have human employees? The simple answer is that financially autonomous software may prove to be useful. A network of vending machines may not *need* a human or organizational owner; it may be able to offer items at lower prices if it need not produce a profit, and it is at least a possibility that the dictates of an algorithm will make a service more competitive, at least in some contexts, than the passions of a business school graduate. Moreover, the notion of this sort of business is not in fact unfamiliar; we might conceive an autonomous business simply as a conventional not-for-profit organization that requires minimal manual oversight.

³¹ Cf. Hearn, *supra* note 29, at 15:18.

³² Cf. SHAWN J. BAYERN, CLOSELY HELD ORGANIZATIONS 93–94 (2014).

If such businesses are to arise, how might they interact with legal organizational structures? One possibility is that their creators aim to sidestep the law and avoid concerning themselves with organizational forms. I leave that problem to the next section, however; for now, consider how a responsible creator of an autonomous system might interact with American organizational law.

As background, there has recently been some debate as to whether modern partnership law, such as the Revised Uniform Partnership Act (RUPA), permits partnerships that have a single “partner” rather than the more typical pattern of multiple parties acting together to co-own a business.³³ The question is primarily one of statutory interpretation. RUPA defines a partnership as “an association of two or more persons to carry on as co-owners a business for profit formed under Section 202, predecessor law, or comparable law of another jurisdiction”³⁴ but then outlines processes by which a partnership may come to have only one partner.³⁵ The statute’s definition of a partnership does not resolve the ambiguity; arguably, it requires only that a partnership have two or more partners at the moment it is formed.³⁶ It would be formalistic, or at least require further reasoning, to suppose that a partnership must terminate because it fails to meet the definitional criteria that govern its formation.

A more nuanced question lurks beneath the surface, however. If a partnership might have a single partner operating without co-owners, why does it need even a single partner? Similarly, and probably more practically, why would a flexible organization like an LLC need to have at least one member at all times? Indeed, a zero-member LLC might arise under relatively mundane circumstances. Consider, for example, an LLC organized to manage a family’s assets; this is a common device for tax, liability, and estate planning.³⁷ Suppose the operating agreement provides that when one parent dies, the other becomes the sole member, and when that sole member dies, those children in the family that meet a certain criterion may elect within thirty days to become managing members of the LLC. It is easily possible that during those thirty days, the LLC may have no members—effectively, no current owners. Indeed, the Revised Uniform Limited Liability Company Act (RULLCA) explicitly provides for that possibility by including, in a list of events that cause the dissolution of an

³³ This debate is played out literally as a dialogue between two of the foremost experts on partnership law in Robert W. Hillman & Donald J. Weidner, *Partners Without Partners: The Legal Status of Single Person Partnerships*, 17 *FORDHAM J. CORP. & FIN. L.* 449, 453–71 (2012).

³⁴ UNIF. P’SHIP ACT § 101(6) (1997).

³⁵ For example, in a two-person partnership, one partner may dissociate under RUPA Section 601(7)(i), and a dissociation of this type will, by default, trigger RUPA’s rules concerning buyout rather than dissolution. *See id.* §§ 601(7), 603, 701, 801(1).

³⁶ *See Hillman & Weidner, supra note 33*, at 459.

³⁷ *See generally* Thomas E. Dew, *Sharing the Family’s Wealth*, 81 *MICH. B. J.* 50 (2002) (describing family limited liability companies).

LLC, “the passage of 90 consecutive days during which the company has no members.”³⁸

But this provision, perhaps surprisingly, appears not to be a mandatory rule imposed by the uniform statute. RULLCA Section 110(c) lists the statute’s mandatory, nonwaivable provisions.³⁹ That list explicitly refers to other criteria that might cause dissolution of an LLC—specifically, to applications by members for court-ordered dissolution as a result of fraud, oppression, or general illegality—but does not refer to the ninety-day window for zero-member LLCs. Perhaps this is an oversight, but following the present version of RULLCA, which several states have adopted,⁴⁰ it appears remarkably straightforward to set up a perpetual LLC that has no members in its final, planned operational state.

The permission of just a single state would be sufficient to enable autonomous businesses. An organizer of such a business merely would need to select the organizational law of a state that permits a perpetual autonomous LLC.⁴¹ And given the connections between organizational law and the rest of the legal system—the notion of legal personality—such an LLC could then enter contracts, hire a lawyer to file lawsuits, own property, set up a bank account, and so on. The full range of Meir Dan-Cohen’s exercise⁴² is then legally possible, and it apparently has become surprisingly straightforward in both law and software.

Of course, the state of artificial intelligence is probably not yet good enough for an autonomous software system to know what it should do if it is sued. But just as autonomous software could hire someone to fill empty vending machines by placing orders with bitcoins, the software that operates an autonomous LLC might hire a lawyer in response to a lawsuit (or in response to a determination that one of its contractual claims is unfulfilled). The software simply needs some authenticable mechanism to know it has received an official communication from a court that has jurisdiction over it. Conceptually, this can be managed using modern cryptography—technology exists for a governmental authority to sign an electronic message in a manner that proves its origin to be authentic⁴³—although in practice, most official communications are not yet issued in a way that would make sense to software.

³⁸ REVISED UNIF. LTD. LIAB. CO. ACT § 701(a)(3) (2006).

³⁹ *Id.* § 110(c).

⁴⁰ See *Limited Liability Company (Revised)*, UNIF. LAW COMM’N, [http://www.uniformlaws.org/Act.aspx?title=Limited Liability Company %28Revised%29](http://www.uniformlaws.org/Act.aspx?title=Limited+Liability+Company+%28Revised%29) (last visited Mar. 30, 2014) (tracking state-by-state adoption of the RULLCA).

⁴¹ *Edgar v. Mite Corp.*, 457 U.S. 624, 645 (1982) (upholding the “internal affairs doctrine” under which an entity is regulated by the organizational law of the state in which it is organized).

⁴² See DAN-COHEN, *supra* note 24, at 46–49.

⁴³ See BRUCE SCHNEIER, *APPLIED CRYPTOGRAPHY* 483–502 (2d ed. 1996) (describing algorithms for digital signatures using public-key cryptography).

That could easily change, however, and indeed it could be tied to the regulation of perpetual LLCs. The state government, in offering perpetual autonomous LLCs, can effectively engage in a quid pro quo with the software designers: We will allow you to create this structure, with the legal properties you expect, but the autonomous LLC must pay a fee every year, must provide a digital mechanism by which legal process can be served, and must satisfy legal judgments against it to continue operating. As it turns out, this is extremely close to how states already treat LLCs; the only difference is that states do not always manage the process online or by means of modern cryptography. For example, it is typical for statutes to require LLCs to register a physical mailing address for service of process.⁴⁴ But the difference between a physical mailing address and email is just a technical detail.

B. Common Law and Regulatory Challenges

If the designers of an autonomous system wish for it to respond responsibly, with the aid of legal counsel, to its torts or breaches of contract, the previous section outlines a surprisingly straightforward mechanism by which that might occur. Just as a conventional LLC might commit torts or breach its contracts and then be held to legal account, so might a zero-member LLC.

The situation is more complicated if the designers opt not to use a zero-member LLC or similarly convenient structure and instead operate without any specific legal structure. In that event, the appropriate legal response becomes murkier. It may not be obvious what the law should do when faced with a software system that has caused, or continues to cause, legally cognizable damage but has no legally responsible organizer. For example, how should the law respond to a software system that has no cognizable owner, legal address, or clear physical presence, yet breaches a supply contract? Perhaps more ominously, how might the law respond to a faceless software system (such as a fanciful package-delivery system based on airborne drones) whose directives have caused physical damage?

A range of responses is possible. For example, the original designer, if located, could be liable as a matter of simple negligence for setting into motion an unreasonably dangerous system. Similarly, those who enable the ongoing operation of a physically dangerous system could be held to have breached the duty of ordinary care to the system's victims. Liability of this kind could extend similarly to breach of contract, because modern law

⁴⁴ *E.g.*, REVISED UNIF. LTD. LIAB. CO. ACT § 113(a) (“A limited liability company shall designate and continuously maintain in this state: (1) an office, which need not be a place of its activity in this state; and (2) an agent for service of process.”).

tends to treat a software program as any other tool by which, for example, a promisor might communicate a contractual offer.⁴⁵

The individual designer's liability probably stands even if he or she chooses to set up an LLC. Too often, people misunderstand the liability-shielding role of an LLC; although it does shield members from vicarious liability as members, it does not exempt individuals from liability for their own torts.⁴⁶ As a result, a selfish software developer may have little reason to go through the trouble of establishing an LLC. Perhaps to encourage the orderly creation of legal structures, however, LLC acts of the future might appropriately treat liability incurred by an electronic agent as vicarious rather than direct, thus providing an incentive for software developers to use LLCs.

Bitcoin itself is probably too decentralized, and too remote for purposes of proximate causation, to provide any targets of liability for enabling an autonomous system. As Part I showed, Bitcoin has no central authority. Even the individual software developers who maintain the current, less-than-fully-decentralized version of the Bitcoin software⁴⁷ probably have an insufficient connection to harms that separately developed autonomous agents might cause. Indeed, it is unclear that such developers would even be a factual cause of the harm for purposes of tort law just because Bitcoin enabled the harm. (The developers' marginal contributions may not themselves have been necessary for the harm to occur.)

Perhaps most interestingly, however, if a judgment could be enforced against an unwilling autonomous system—for example, by invading the system on which it runs and seizing private cryptographic keys—I see no reason that an informally organized autonomous system should not be sued in its own right. That possibility seems conceptually strange, but the procedural rules should aim to achieve functional goals; it would be unhelpful to ignore useful possibilities on formalistic grounds merely because they are conceptually strange. Of course, we would not think it strange to attempt to interdict a dangerous autonomous system as a matter of public safety. Similarly, common law judges should consider themselves capable of such interdiction in private law matters as well; their equitable powers are likely sufficient to issue orders against an autonomous system that has caused damage.

⁴⁵ See RESTATEMENT (THIRD) OF AGENCY LAW § 1.04 cmt. e, illus. 3 (2006) (“At present, computer programs are instrumentalities of the persons who use them. If a program malfunctions, . . . the legal consequences for the person who uses it are no different than the consequences stemming from the malfunction of any other type of instrumentality.”).

⁴⁶ See BAYERN, *supra* note 32, at 245.

⁴⁷ See *supra* notes 18–19 and accompanying text.

Whatever the common law response, the combination of Bitcoin and the zero-member LLC provides at least the possibility of a significantly new type of economic and legal entity.