**TESTIMONY OF**

**Peter Van Valkenburgh**[1]

**Director of Research at Coin Center**

**BEFORE THE**

**United States Senate Banking Committee**

**"Exploring the Cryptocurrency and Blockchain Ecosystem"**

**October 11, 2018**

## Executive Summary

**You may have heard that "blockchain technology" is the solution to any number of social, economic, organizational, or cybersecurity problems. *It is not*. A blockchain is merely a data structure and "blockchain technology" is a vague and undefined buzzword. In this paper, we explain the true technologies that undergird blockchain networks and the distinctions between public and private blockchain networks, why they matter, and why only public blockchain networks can solve certain specific issues related to electronic cash, identity, and the Internet of Things.**

**"Blockchain technology" is not a helpful phrase.** It abstracts real, specific technical innovations into a generalized panacea. The phrase suggests a vague design pattern, which is then trumpeted as the solution to all manner of societal and organizational problems. And amongst all of this cheerleading, almost nothing is ever offered in the way of real design specifics. This tends to be because **"blockchain technology" is described monolithically**, as if there are no specific design choices to be made in building "blockchain solutions" beyond choosing to use a blockchain. The advantages and disadvantages of various approaches and technical architectures are generally not discussed (except perhaps by experts) and the non-technical public is left with a warm blanket and little understanding of why any of this matters.

**This testimony offers specifics.** It begins by describing **why "decentralized computing" matters**. If all of the "blockchain technology" hype has one thing in common, it's the idea that ***a computer application, which creates some useful result for its users, can be run***

[1] Peter is Director of Research at Coin Center, the leading independent non-profit research and advocacy group focused on the public policy issues facing cryptocurrency technologies such as Bitcoin. This testimony is based largely on a report published by Coin Center. *See* Peter Van Valkenburgh, "Open Matters: Why Permissionless Blockchains are Essential to the Future of the Internet" *Coin Center* (2016) https://coincenter.org/entry/open-matters.

***simultaneously on many computers around the world rather than on just one central server, and that the network of computers can work together to run the application in a way that avoids trusting the honesty or integrity of any one computer or its administrators.*** To describe this idea we prefer the term "decentralized computing" to "blockchain technology," because it is more descriptive and it is also a broader category.

**This testimony demystifies the actual technologies behind "blockchain technology"** and explains these *several* technologies in a way that even non-technical readers will understand. This testimony creates a typology of "blockchain technologies" and it will suggest that only certain *types* of "blockchain technology" can be real solutions to certain major social and organizational challenges.

For starters, rather than talking about "blockchain technology" in the abstract, we discuss the real technical innovations that underlie Bitcoin, the actual functioning technology that has spurred all the blockchain hype. There are really **three core innovations** that underlie Bitcoin: **peer-to-peer networking**, **blockchains**, and **consensus mechanisms**. Of these, peer-to-peer networking is generally nothing new, and blockchains are merely novel ways of storing and validating data. ***Consensus mechanisms, however, are the truly disruptive, interesting, and critical component of the design.*** **When it comes to capabilities, risks, and disruptive potential, however, not all consensus mechanisms are created equal.** The critical nature of consensus mechanisms in these new blockchain-powered decentralized computing systems, and the variability in types of consensus mechanism design are why the bulk of this testimony focuses on explaining consensus mechanisms to non-technical audiences.

In general, **by consensus we simply mean the process by which a number of computers come to agree on some shared set of data and continually record valid changes to that data.** So the blockchain might be the form that the data take, *e.g.* a hashed list of valid transactions in bitcoin, but it is the consensus mechanism that generates that blockchain, validates the data, and continually keeps the data updated and reconciled between all of the computers in the system.

This brings us to the question of "publicness" in the consensus mechanism. Who is allowed to read the data over which the network is forming consensus, and possibly more important, who is allowed to participate in the process that ultimately results in new data being added? Are some consensus mechanisms more open to free participation than others? **In a public consensus mechanism anyone with a computer and an internet connection should be eligible to play a role in writing consensus data; in a private consensus mechanism only those who have been identified by a centralized authority and given an authorization credential are allowed to participate.**

The operation of various consensus mechanisms is described in the full testimony. Public consensus mechanisms include **proof-of-work** based mechanisms, as found in Bitcoin and most cryptocurrencies, as well as **proof-of-stake** mechanisms and **social consensus** mechanisms. Private consensus mechanisms generally follow what we call a **consortium consensus** model, wherein only identified and credentialed consortium members share the privilege of writing consensus data.

From an **innovation policy** perspective, public consensus mechanisms are superior to their

private counterparts because they create purpose-agnostic platforms atop which anyone with a connected computer can build, test, and run user-facing decentralized applications. In this sense, **networks powered by public consensus mechanisms mirror the early Internet, and may one day become as indispensable as the Internet in facilitating free speech, competition, and innovation in computing services.**

Apart from publicness, we also discuss the nature of **trust** and **privacy** in each of the several consensus mechanisms. Public consensus mechanisms demand that users place trust in unknown third parties who are economically motivated to behave honestly because they have **skin in the game** and face **competitive pressures**. Private consensus mechanisms demand that users place trust in the identifying authority who provisions consortium members with credentials, and the honesty and cybersecurity practices of the members themselves. Public consensus mechanisms trade **transparency** for **privacy** but new technologies such as **zero-knowledge proofs** and **homomorphic encryption** may enable public networks to have superior privacy and verifiability as compared with private networks that rely only on **perimeter security** to maintain privacy.

Finally, we explain why public consensus mechanisms, specifically, are critical for three particular decentralized computing applications: **electronic cash, identity, and the Internet of Things**.

- **Electronic Cash.** Truly electronic ***cash*** (*i.e.* fungible bearer assets, the use of which resembles that of paper notes) offers **efficiencies that existing electronic money transmission systems cannot**. There are hidden costs to legacy systems: chargebacks, and transactions forgone because fees are greater than the value being sent or because participants cannot obtain a banking relationship. Fundamentally, from a user's perspective, a private-blockchain money transmission technology doesn't "just work" from the get-go. I cannot send or receive money until I open an account and establish a legal relationship with a company. This may be a tolerable inconvenience, but it is not a system that works like cash, which can be accepted in the hand without any prior arrangements in place. **Only public consensus mechanisms, by fully automating the creation and maintenance of a ledger according to pre-established rules and economic incentives, can offer electronic transactions that are as good as cash.**

- **Identity. The Internet lacks a native identity layer.** This shortcoming is the reason why Internet users must rely on a tapestry of weak passwords, secret questions, and knowledge of mothers' maiden names to verify their identity to various web service providers. The need for a better solution is widely recognized, and **by creating a shared and unowned platform for recording identity data, public blockchains may provide the answer.**

- **The Internet of Things.** Firstly, public blockchain networks allow for a truly decentralized data structure for device identity (I am a bulb in this home's kitchen) and user access authorization (the user with address 0xE1A… is the only person who can turn me on and off). The redundant and decentralized nature of data on these networks can ensure that these systems have true longevity, and that **a manufacturer's decision to end support for a product will not destroy the user's ability to securely access the product's features**.

Secondly, **public blockchain networks can help ensure that devices are interoperable and compatible** because critical infrastructure for device communication, data storage, and computation can be commoditized and shared over a peer-to-peer network rather than be owned (as a server warehouse is owned) by a device manufacturer that may be reticent to opening its costly platform to competitors.

Lastly, **device payments for supporting and maintaining that networked infrastructure or allowing the device's user to easily engage in online commerce can be made efficient** by utilizing the electronic cash systems that only public consensus mechanisms can facilitate.

A public consensus mechanism decentralizes trust, spreading out power on the network across a larger array of participants. For any use-case, this decentralization helps ensure **user sovereignty**, **interoperability**, **longevity**, **fidelity**, **availability**, **privacy**, and **political neutrality**. In the full testimony, the necessity of these attributes is explained in the context of each decentralized computing application (electronic cash, identity, and the Internet of Things), and a discussion of public and private consensus mechanisms for that application follows.

# Contents


Executive Summary 1

I. The Decentralized Computing Revolution 6

A. An Easy Introduction to Decentralized Computing 6

B. Platforms for Innovation: Computing, Sharing, Trusting 9

C. Platforms for Innovation: Public or Private 11

D. The Internet and Permission 13

II. Making Sense of Consensus 15

A. Proof-of-Work 17

B. Proof-of-Stake 21

C. Consortium Consensus 22

D. Social Consensus 23

III. Publicness, Trust, and Privacy Across Various Consensus Models 23

A. Publicness Across Consensus Mechanism 24

B. Trust Across Consensus Mechanisms 28

C. Privacy Across Consensus Mechanisms 35

IV. Use Cases in which Public Consensus is Critical 44

A. Electronic Cash 45

B. Identity 52

C. The Internet of Things 58

V. Conclusion 67

## I. The Decentralized Computing Revolution

If all of the "blockchain technology" hype has one thing in common, it's the idea that a computer application, which creates some useful result for its users, can be run simultaneously on many computers around the world rather than on just one central server, and that the network of computers can work together to run the application in a way that avoids trusting the honesty or integrity of any one computer or its administrators. To describe this idea, we prefer the term "decentralized computing" to "blockchain technology," because it is more descriptive and it is also a broader category.

### A. An Easy Introduction to Decentralized Computing

The easiest way to understand decentralized computing is to begin by thinking about a computer program you use and with which you are comfortable. It could be any computer program that you use for work or for fun. For this example, let's just pick a *word processor*. Sure it's not the most titillating software out there, but pretty much everyone who has ever used a computer has used a word processor at some point in their digital lives.

Let's think about the history of the word processor. In the *old* days—the 1990s no less—word processing, like dying, was something you always did alone. If you used Microsoft Word, Wordperfect, or MacWrite, you were running software that used *only* the processor, memory, disk space, monitor, and keyboard of *your personal computer*. The word processor was software trapped on an island. If you wanted to share your draft for the next great American novel, then you would either need to print it or save it as a file on a disk and hope your editor, reader, or critic had the same word processing software as you and could open the file on her own island-like computer. If she made edits she would need to send the file back and you would need to merge her changes with any changes you had made since she got a copy. Frustrating, but a real improvement over piles of redlined paper.

Fast forward to the 21st century and new word processing applications began to make collaboration easier, most notably Google Docs and Microsoft Word with OneDrive. These new services took advantage of what marketing executives persuasively and reassuringly dubbed "the cloud." Word processing via the cloud means it is much easier to work with others in creating a document; in the best implementations you can control who has read or write access, see your co-authors typing in real time, comment and discuss changes, and see a full history of everyone's edits.

From a computing standpoint this is not cloud magic. What is really happening is that the word processor software is no longer running on your island-like computer; it is running on a server that Google or Microsoft owns and maintains somewhere in a giant warehouse somewhere in the world. The interface that we see on our computers when we use these services is just that, an interface—a way to communicate with the computer that Google or some other cloud services provider owns and controls. Collaboration is a cinch with these systems because every editor can have an interface that talks to the same central computer.

The software is still running on an island, but it's an island that everyone can connect to.

Decentralized computing systems now under development present a new opportunity. Rather than moving the computation from the user's device to a centralized server in order to facilitate collaborative applications like Google Docs, we could instead replicate the computation across the otherwise island-like computers of all users.

Imagine I've got an idea for the next hit young adult novel about dragons, and I have a co-author/by-day-herpetologist who is great at describing the scales, a cold-blooded editor at Penguin who is ready to viciously rip apart our draft, and a family of dragon-enthusiast sons, daughters, nieces, and nephews who are the ideal focus group for dragonian feedback. How can we all work together to get this dragon tale off the ground? Rather than all of us connecting to a central server to view and edit the shared draft, we could have all our computers connect to each other in a decentralized web, and our computers could work together to agree upon, and stay in sync with, the latest draft, edits, discussions, and permissions describing who is allowed to edit, comment, or read.

That is decentralized computing: the ability to run applications not on your own island-computer or on someone else's central computer, but on a truly nebulous cloud computer not owned or controlled by any single party.

Our word processing example has now, however, reached the end of its usefulness. As the PC and the Internet proved, it is not a single application like word processing that forges the value of today's information superhighway. The value is in the highway itself: a general purpose computing platform, full of cars, buses, vehicles of all types and colors helping people reach all sorts of destinations.  As discussed in the next section, the development of these purpose-agnostic platforms is the true decentralized computing revolution at hand.

### B. Platforms for Innovation: Computing, Sharing, Trusting

The PC and the Internet were revolutionary not because they were self-contained innovations, but rather because they were platforms for innovation. Decentralized computing tools like Bitcoin and Ethereum, discussed throughout, are the beginnings of a new platform for innovation that promises to facilitate a third wave of computing. The PC gave us home computing and productivity applications; the Internet gave us networked computing, collaboration, and rich audio-visual communication; and decentralized computing will give us tools to enable trust, exchange, and community governance.

The PC enabled a wave of consumer and professional applications, from word processing to gaming, from music production to 3D design. Abruptly, the child of a middle income household had a printing press, a cavernous arcade, a recording studio, a suite of architectural drafting tools and paper, and more at her fingertips in a box that sat inconspicuously in her parents' home office.

Then the Internet allowed these otherwise isolated productivity tools to be networked, to

speak to the world. The PC ran applications, and the Internet enabled those applications to communicate globally, to be multi-user, to share data. Now the home printing press was matched with a fleet of newspaper delivery trucks; the arcade, still cavernous, was open to players across the world who could compete with each other; the recording studio came with a record label, trucks to ship vinyl, and stores to sell hits; the architectural tools came with virtual warehouses of objects, furniture, homes, and vehicles waiting to be built or even printed in 3D.

The Internet created a uniform mechanism for computers to speak to each other, but it did not create a uniform mechanism for verifiable agreement (what we might call "trust") between two or more computers and their two or more users. As cryptographer Nick Szabo has written:

> When we currently use a smartphone or a laptop on a cell network or the Internet, the other end of these interactions typically run on other solo computers, such as web servers. Practically all of these machines have architectures that were designed to be controlled by a single person or a hierarchy of people who know and trust each other. From the point of view of a remote web or app user, these architectures are based on full trust in an unknown "root" administrator, who can control everything that happens on the server: they can read, alter, delete, or block any data on that computer at will.[2]

We have come to call shared computing tools "cloud computing," but, marketing aside, *there is no cloud, there's just other people's computers.* So when, today, we engage in any sort of shared computing—whether it be social networking, collaborative document editing, shopping, online banking, or posting a video of our pets—we are utilizing the computers of an intermediary—whether it be Facebook, Google, Amazon, Bank of America, or YouTube respectively. Those intermediaries have control over everything that happens on their servers. They can see a wealth of our personal data and users trust them to only use and manipulate that data according to user instructions and in the best interest of users. Any agreement or level of trust between two users of a given intermediary's service—as when I sell my car to another eBay user, or recognize the positive eBay feedback and reputation of the prospective buyer—is established and maintained by that intermediary.

This architecture has been essential to the rise of the Internet and collectively we have benefited tremendously from the creation of these shared computing systems. It does, however, introduce a great deal of trust into consumer-business relationships; trust that can be misplaced and abused if an intermediary maliciously misuses their customer's data, fails to

---

[2] Nick Szabo, "The dawn of trustworthy computing" *Unenumerated* (Dec. 2014) http://unenumerated.blogspot.com/2014/12/the-dawn-of-trustworthy-computing.html. *See also* IBM Institute for Business Value, *Device Democracy: Saving the future of the Internet of Things* https://public.dhe.ibm.com/common/ssi/ecm/gb/en/gbe03620usen/GBE03620USEN.PDF ("The Internet was originally built on trust. In the post-Snowden era, it is evident that trust in the Internet is over. The notion of IoT solutions built as centralized systems with trusted partners is now something of a fantasy. Most solutions today provide the ability for centralized authorities, whether governments, manufacturers or service providers to gain unauthorized access to and control devices by collecting and analyzing user data.").

secure it from hackers, or profits unfairly from a user who is locked into the service and finds it difficult to migrate their data to a competing service provider.

New and emerging computing architectures can help forge trustworthy relationships directly between users without intermediaries. The most visible of these new systems thus far is Bitcoin, a peer-to-peer network protocol that allows users to hold and send provably scarce tokens (bitcoins) that can function like cash for the Internet. Electronic cash, however, is just one potential computing service that can be designed to be intermediary-less, to run across the computers of a decentralized network of users rather than on the centralized servers of a particular service provider.

At root, any shared computing system can be thought of as a single shared computer, a computer made up of computers. Bitcoin is, following this logic, a computer made up of many computers whose several users have installed and are running Bitcoin-compatible software. Working together, all of these computers periodically come to an agreement over the ledger of all Bitcoin transactions—the Bitcoin blockchain. That ledger is, at any moment, the authoritative "state" of the decentralized Bitcoin computer. But computer "state" can be any data, not just a list of cash-like transactions. For example, when using Microsoft Word, a writer is perpetually updating the state of her computer, typing word after word into a document whose current changes—the current state—continually appear on the screen.

If a decentralized network of computers can continuously agree on the most recent and updated state of all interactions on that network—like keystrokes to a Word document—then it could be programmed to perform the computations necessary for any number of applications. Tracking the reputation of sellers and buyers, permissioning editing or access rights to a shared document, rewarding creative contributors for popular video content, any of the previously described "cloud" services provided by intermediaries could be programmed into a decentralized computing network. As Szabo has noted,

> Much as pocket calculators pioneered an early era of limited personal computing before the dawn of the general-purpose personal computer, Bitcoin has pioneered the field of trustworthy computing with a partial block chain computer. Bitcoin has implemented a currency in which someone in Zimbabwe can pay somebody in Albania without any dependence on local institutions, and can do a number of other interesting trust-minimized operations, including multiple signature authority. But the limits of Bitcoin's language and its tiny memory mean it can't be used for most other fiduciary applications[.][3]

Several efforts are underway to design systems that can enable a larger range of "fiduciary" applications, systems that will be effectively *general purpose decentralized computers*: platforms for trustworthy shared computing just as flexible and repurposable as the PC and the Internet have become. Some of these systems modify or build on top of Bitcoin (Rootstock[4] and

[3] *Id.*
[4] Sergio Demian Lerner, *RSK Rootstock Platform: Bitcoin Powered Smart Contracts* (Nov. 2015)

Blockstack[5] among others), others are new standalone network protocols (the largest by value is Ethereum[6]). Still others are building decentralized computing systems that are private or permissioned by default (most notably Corda by R3CEV[7]), in order to allow a pre-specified set of users to agree upon some limited-purpose computation—like validating contracts between banks.

The component parts of these new architectures are generally three-fold: peer-to-peer networking, blockchains, and consensus mechanisms. All three of these concepts are often lumped together under the general and impressive-sounding heading "blockchain technology," but for clarity this testimony will deal with each separately and will ultimately focus on the third lump—consensus mechanisms—because it is the architecture of this third component that has the most important implications for building useful and well-functioning decentralized applications.

You can think of these three technologies as follows: *peer-to-peer networking* is how connected machines communicate with each other, *blockchains* are the data structures the connected peers use to store important variables in the shared computation, and the *consensus mechanism* is the tool to generate the shared and agreed-upon computation itself.

As we will discuss, the architecture of the consensus mechanism is important to consider. Different choices may have different outcomes for users—more or less privacy, more or less choice, more or less costs to participation. Just as the fundamental technical architecture of the PC and the Internet had long-term ramifications for the relative fairness, distribution and availability of computing and communication tools, so may choices in the now-unfolding architecture of consensus.

As we will explain, *all* new approaches to decentralized computing—whether private or public—should be celebrated and allowed to develop relatively unfettered by regulatory or government policy choices much as the Clinton Administration took a light-touch approach to the development of the Internet in the 1990s.[8] In order to make those choices, however,

---

https://uploads.strikinglycdn.com/files/90847694-70f0-4668-ba7f-dd0c6b0b00a1/RootstockWhitePaperv9-Overview.pdf

[5] Muneeb Ali, Jude Nelson, Ryan Shea and Michael J. Freedman, *Blockstack: A Global Naming and Storage System Secured by Blockchains* (June 2016) https://blockstack.org/blockstack.pdf

[6] Vitalik Buterin, *Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform* (Jan. 2014) https://github.com/ethereum/wiki/wiki/White-Paper

[7] Richard Gendal Brown, James Carlyle, Ian Grigg, Mike Hearn, *Corda: An Introduction* (Aug. 2016) https://static1.squarespace.com/static/55f73743e4b051cfcc0b02cf/t/57bda2fdebbd1acc9c0309b2/1472045822585/corda-introductory-whitepaper-final.pdf

[8] President William J. Clinton, Vice President Albert Gore, Jr. *A Framework For Global Electronic Commerce* (July 1997) *available at* https://www.w3.org/TR/NOTE-framework-970706#Annotated Version ("Governments can have a profound effect on the growth of commerce on the Internet. By their actions, they can facilitate electronic trade or inhibit it. Knowing when to act and -- at least as important -- when not to act, will be crucial to the development of electronic commerce.5 This report articulates the Administration's vision for the emergence of the GII as a vibrant global marketplace by suggesting a set of principles, presenting a series of policies, and establishing a road map for international discussions and agreements to facilitate the growth of commerce on the Internet.")

policymakers need a basic understanding of how consensus works and what it might help us build.

### C. Platforms for Innovation: Public or Private

A fundamental question in the design of any consensus mechanism is who can participate and how do they participate in order to reach consensus over some shared computation. For many years it was assumed that useful consensus mechanisms could only be developed if the participant computers were identified through channels outside of the decentralized computing system itself.[9] In other words, it had been assumed that useful consensus mechanisms could only be designed as private or permissioned systems: to participate in the decentralized computing system a user would need to either (a) gain physical access to a private underlying network architecture (*e.g.*, an "intranet" rather than the Internet) or (b) obtain an access credential via a cryptographic key exchange with other participants or by utilizing a public key infrastructure.[10] Several such private consensus mechanisms have been, and are continuing to be, developed.[11]

Private consensus mechanisms, however, may not be optimal for the development of robust

---

[9] *See* Jonathan Katz, Andrew Miller, and Elaine Shi, "Pseudonymous Broadcast and Secure Computation from Cryptographic Puzzles" (Oct 2014) *available at* http://eprint.iacr.org/2014/857.pdf ("Standard models of distributed computing assume authenticated point-to point channels between parties, where authentication may be provided via some physical property of the underlying network or using keys shared by the parties in advance. When security against a large fraction of corruptions is desired, even stronger pre-existing setup—*e.g.*, a broadcast channel or a public-key infrastructure (PKI) with which broadcast can be implemented—is often assumed. Such setup may not exist in many interesting scenarios, especially open, peer-to-peer networks in which parties do not necessarily have any prior relationships, and can come and go as they please. Nevertheless, such setup is often assumed due to the prevailing belief that nothing "interesting" can be achieved without them, and in fact there are known impossibility results to this effect."). *See also* Boaz Barak, Ran Canetti, Yehuda Lindell, Rafael Pass, and Tal Rabin. "Secure computation without authentication." *Advances in Cryptology—CRYPTO 2005*, pp. 361–377 (2005).

[10] *Id.*

[11] *See*, *e.g.*, Paxos, a widely used protocol for generating consensus across a set of unreliable processors. Marshall Pease, Robert Shostak, and Leslie Lamport, "Reaching Agreement in the Presence of Faults," 27 *Journal of the Association for Computing Machinery* 228–234 (April 1980). We will not discuss Paxos or related consensus mechanisms within this paper. These systems are generally fault tolerant only under an assumption that none of the nodes are actively attempting to undermine the consensus by sending malicious and deceptive data to other nodes. The ability to deliver a useful distributed computing service despite the presence of malicious and deceptive participants is known in computer science as "byzantine fault tolerance" or BFT. *See* Kevin Driscoll, Brendan Hall, et a*l*, "Byzantine Fault Tolerance, from Theory to Reality" 2788 Lecture Notes in Computer Science 235 (2003). There are BFT variants of Paxos, however, they do not scale effectively to large, highly distributed computing networks. *See* Marko Vukolic, "The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication," *IBM Research* ("This is true even for their crash-tolerant counterparts, i.e., replication protocols such as Paxos, Zab and Raft, which are used in many large scale systems but practically never across more than a handful of replicas."). Accordingly, Paxos is a useful tool for generating an agreement amongst several computers all under one individual or institution's control. The technologies discussed in this paper are limited to newer mechanisms, inspired by Bitcoin, that seek explicitly to generate agreement amongst a large number of computers controlled by mutually distrustful strangers.

general purpose decentralized computing systems. Access to dedicated network infrastructure and/or public key infrastructure is costly, potentially limiting participation to larger players like businesses. In some cases, these prerequisites are irreconcilable with the desired decentralized computing use case, as when consensus is sought across a peer-to-peer network that allows peers free entry and exit.[12] If, as described in the previous section, we believe that some decentralized computing systems should be public platforms for democratic and diverse innovation (as were the PC and the Internet), then a permissioned system seems like a poor choice.

Private systems may be the smarter choice for limited rather than general purpose decentralized computing tasks, where consensus need not be open to all potential participants and participants can be centrally identified and trusted not to collude against the interests of the group (*e.g.*, when a consortium of banks wants to settle inter-bank loans according to a decentralized ledger).[13] Permissionless systems are arguably more difficult to scale,[14] to make private,[15] or to secure than private systems.[16] These, however, are technical challenges that may prove to be fully surmountable.

Much of the current skepticism exhibited by proponents of simpler, private systems could prove shortsighted. Similar issues of scale and usability clouded early predictions about computing generally. For example, in 1951 Cambridge mathematician Douglas Hartree suggested that "all the calculations that would ever be needed in [the UK] could be done on three digital computers—one in Cambridge, one in Teddington, and one in Manchester. No one else would ever need machines of their own, or would be able to afford to buy them."[17] Similar skepticism stalked the early Internet. For example, in 1998 economist Paul Krugman wrote,

> The growth of the Internet will slow drastically, as the flaw in "Metcalfe's law"–which states that the number of potential connections in a network is proportional to the square of the number of participants–becomes apparent: most people have nothing to

[12] Katz, *supra* note 9.
[13] *See*, *e.g.*, Gendal Brown, *supra* note 7.
[14] *See* Vukolic, *supra* note 11. *See also* Kyle Torpey, "Bitcoin Reaches a Crossroads With the Scaling Debate, Not a Crisis" *Bitcoin Magazine* (May 2016) https://bitcoinmagazine.com/articles/bitcoin-reaches-a-crossroads-with-the-scaling-debate-not-a-crisis-1462980183.
[15] *See infra* p. 35.
[16] *See* Robert Sams, "No, Bitcoin is not the future of securities settlement," (2015) http://www.clearmatics.com/2015/05/no-bitcoin-is-not-the-future-of-securities-settlement ("If you are prepared to use trusted third parties for authentication of the counterparts to a transaction, I can see no compelling reason for not also requiring identity authentication of the transaction validators as well. By doing that, you can ditch the gross inefficiencies of proof-of-work and use a consensus algorithm of the one-node-one-vote variety instead that is … thousands of times more efficient.").
[17] Lord Bowden, 58 *American Scientist* 43 (1970). This accurate quotation is generally considered to be the basis for a notorious misquote of IBM President Thomas J Watson, "I think there is a world market for maybe five computers." Brader, Mark (July 10, 1985). "Only 3 computers will be needed..." (Forum post). https://groups.google.com/forum/#!msg/net.misc/390t08t_SZY/d2uJwCwcyQAJ.

> say to each other! By 2005 or so, it will become clear that the Internet's impact on the economy has been no greater than the fax machine's.[18]

The development of the Internet defied many such skeptics. Before we discuss exactly how public and private consensus mechanisms work, it's important to understand how the internet was and is itself *public*, and how that publicness proved essential to its success.

### D. The Internet and Permission

The Internet is revolutionary in large part because it avoids the costs of permissioning described above. The underlying protocols that power the Internet—TCP/IP (the Transmission Control Protocol and the Internet Protocol)—are open technical specifications.[19] Think of them like human languages; anyone is free to learn them, and if you learn a language well you can write anything in that language and share it: books, magazines, movie scripts, political speeches, and more. Importantly, you never need to seek permission from the *Institut Français* or the *Agenzia Italiana* to build these higher level creations on top of the lower level languages. Indeed, no one can stop you from learning and using a language.

When Tim Berners Lee had the idea of sending virtual pages filled with styled text, images, and interactive links over TCP/IP (*i.e.* when he invented the Word Wide Web),[20] there was no central authority he needed to approve the project. He could write the standards and protocols for displaying websites—the higher level internet protocol known as HTTP (the HyperText Transfer Protocol), and anyone with a TCP/IP capable server or client could run freely available HTTP-based software (web-browsers and web-servers) to read or publish these new rich web pages.[21] As a result, the Internet went from a primarily command-line text-only interface to a virtual magazine full of pleasantly styled pages full of text, pictures, and links to other related pages, and it made the transition without any formal body approving the change. Every Internet user was free to opt in or opt out of the new format, the World Wide Web, as they so desired simply by choosing whether or not to read and write internet data with the new higher level protocol, HTTP.

Today, thanks to the public, permissionless architecture of TCP/IP and higher level protocols built on top of it, no one needs to gain access to a private network in order to create a blog or send an email. Nor must an Internet user obtain a certificate of identity to participate in online discussions. Nor must a hardware designer obtain permission to build a new gadget that

---

[18] Megan Mcardle, "Predictions are Hard Especially About the Future" *The Atlantic* (Dec. 2010) http://www.theatlantic.com/business/archive/2010/12/predictions-are-hard-especially-about-the-future/68471/.
[19] Lydia Parziale, *et al.*, *TCP/IP Tutorial and Technical Overview* (Dec. 2006) *available at* https://www.redbooks.ibm.com/redbooks/pdfs/gg243376.pdf.
[20] World Wide Web Foundation, *History of the Web*, http://webfoundation.org/about/vision/history-of-the-web/ *last accessed* Dec. 2016 ("Had the technology been proprietary, and in my total control, it would probably not have taken off. You can't propose that something be a universal space and at the same time keep control of it.").
[21] *Id.*

can send and receive data from the Internet.[22] This publicness has been a major factor in democratizing communications, and spurring vibrant competition and innovation. Anyone can design, build, and utilize hardware or software that will automatically connect to the Internet without seeking permission from a network gatekeeper, a national government, or a competitor.

It is true that businesses often utilize public key infrastructure online, and that this does add a layer of permissioning to the web. When you visit an online bank, for example, your web browser will look for a signed certificate issued by a *certificate authority* that has vouched for the bank's online identity.[23] This begins a process between your browser and the bank that will ultimately encrypt all of your communications while you are navigating the website. This process is known as TLS/SSL (Transport Layer Security and its predecessor, Secure Sockets Layer), and it is the system behind the little green lock consumers are told to watch out for when visiting sensitive websites like banks.[24]

TLS/SSL, however, is another application-layer Internet protocol—like HTTP—that runs *on top* of the public TCP/IP network. Again, the underlying protocols are the reason for the Internet's publicness. When a consumer device is connected to the Internet these protocols do not ask for identification, certificates, or authentication; they simply assign the new device a seemingly random but unique pseudonym (called an IP Address) in order to have a consistent address for routing data.[25] The identified and permissioned layer, TLS/SSL, is running on top of the public and pseudonymous layer.

The layered design of the Internet is not accidental. It is modular, with a public lower layer, in order to enable flexibility. One can always build identified and permissioned layers on top of a permissionless system—as TLS/SSL (a private, identified layer) is built on top of TCP/IP (a public, pseudonymous layer). The reverse is not possible, however. Had the Internet originally been architected to be permissioned and identified, it would have imposed costs and limitations on public participation, and it would have ossified the possible range and diversity of future higher level protocols for identity and permission. When lower layers are permissionless and pseudonymous, on the other hand, the costs of participating are low (merely the cost of hardware and free Internet-protocol-ready software), and such a open platform enables a variety of private or identified higher level layers to emerge and compete for particular use cases where identity and permissioning are essential. For example, PGP and the Web of Trust compete with TLS/SSL as methods for enabling secure and identified

---

[22] *Id. See also* W3C, *Web of Devices* https://www.w3.org/standards/webofdevices/ *last accessed* Dec. 2016. ("W3C is focusing on technologies to enable Web access anywhere, anytime, using any device. This includes Web access from mobile phones and other mobile devices as well as use of Web technology in consumer electronics, printers, interactive television, and even automobiles.").
[23] Microsoft, *What is TLS/SSL?* (Mar. 2003) https://technet.microsoft.com/en-us/library/cc784450(v=ws.10).aspx.
[24] Google, *Check Chrome's connection to a site* https://support.google.com/chrome/answer/95617?hl=en *last accessed* Dec. 2016.
[25] *See* Stephanie Crawford, "What is an IP address?" *How Stuff Works* http://computer.howstuffworks.com/internet/basics/question549.htm *last accessed* Dec. 2016.

communications built on top of TCP/IP.

We are still in the very early days of decentralized computing systems, and there remains much uncertainty over which protocols and systems will come to dominate the space. Given that uncertainty, it is possible that these systems will not follow the evolution of the Internet or the PC and instead be permissioned by default at the lower level. The key takeaway from a policy perspective, however, should be (1) awareness of the technological features that enabled the Internet to flourish as a democratic and innovative medium—modularity, publicness, and pseudonymity; and (2) a willingness to allow these new decentralized computing systems to evolve similarly unencumbered even when publicness and pseudonymity cause regulatory confusion or concern because of their newness and sharp contrast with legacy systems.

## II. Making Sense of Consensus

It's easy to be excited about the *applications* that can be built on top of decentralized computing platforms. They usually have an easy and provocative elevator pitch: *this app will let you send money instantly*, and *this app will save you from creating and remembering hundreds of passwords!* Talking about the infrastructure that powers and enables those apps, however, is harder because the discussion will often be laden with technical jargon and the purpose of the system will be more abstract (*i.e.*, to create a platform for applications that have human-facing purposes).

These underlying architectures, however, have real ramifications for consumer protection and freedom of choice, so it's important that policymakers and concerned citizens understand the various models that are being developed. Just as it can be daunting to learn about internal combustion or gene sequencing, we understand that knowledge of these topics is key to forming good policy for car safety or GMO foods. Similarly, policy aimed at regulating the application level of decentralized computing (*e.g.*, money transmission, identity provision, consumer device privacy) should be informed by knowledge of the underlying infrastructures. This section will explain those technologies in general, but first a disclaimer:

This is not a document intended for technologists, and many of the salient features of these mechanisms will be spoken of in the abstract. Just as one can explain the principles behind internal combustion engines without discussing the acceptable tolerances in the machining of a piston and gudgeon pin, we will attempt to give an accurate general description of decentralized consensus while avoiding discussion of the merits of sharding or SHA-256.

Speaking generally, the goal of a consensus mechanism is to help several networked participant computers come to an agreement over **(1)** ***some set of data,*** **(2)** ***modifications to or computations with that data, and*** **(3)** ***the rules that govern that data storage and computation.***

To use Bitcoin as an example, the network of Bitcoin users run software with an in-built consensus mechanism. This consensus mechanism helps all of the peers on the network

(Bitcoin users):

1. ***Store agreed-upon data:*** every peer gets a copy of the full ledger of all bitcoin transactions in the history of the network.
2. ***Compute and transform that data:*** recipients of bitcoin transactions can write new transactions thus adding to the ledger all transactions.
3. ***Agree on rules for how storage and computation of that data can take place:*** the ledger is continually updated because all peers listen for and relay new transactions if they are valid, and a lottery is used to periodically pick a random peer to state the authoritative order of valid transactions for chunks of time that are about 10 minutes long. (There are other rules but these are probably the most general and fundamental Bitcoin consensus rules).

If this example is not entirely clear, that's OK. We will expand upon it later in this testimony. The key thing to remember is that *consensus* means that a network of peers can agree upon three things: **(1)** ***data,*** **(2)** ***computation (transformation of the data), and*** **(3)** ***the rules for how computation can take place***.

Any particular *consensus* mechanism can be designed to leverage two techniques in order to ensure agreement over a computation and the associated data.

First, there are what we can call ***automatic rules***. To use an automatic rule, all parties to the consensus can run software on their computers that automatically rejects certain "invalid" computational operations or outcomes on sight. To make a legal analogy, we can think of this as *res ipsa loquitur* (the principle that the mere occurrence of an accident implies negligence), or a rule of strict liability.

For example, Bitcoin's core software defines certain outcomes as always impermissible on sight. Most notably, transactions from one user to another cannot send any bitcoins that have not previously been sent to the sender.[26] More simply: I can't hand you cash that hasn't previously been given to me. To be compatible with the larger Bitcoin network, the software you run on your computer *must follow this rule*. If it does not, other nodes on the network will ignore any invalid messages you send using it. You can try to send the network messages that attempt such counterfeiting, but your messages will always fall on deaf ears and the effort will be futile. These are automatic rules that help the network ignore data that is irrelevant or malevolent to the agreement the participants are seeking.

Second, there are what we can call ***decision rules***. In situations where there are two differing outcomes from the computation, but where both would be valid based on the automatic rules, a rule of decision between each possible valid state is needed in order to keep the network in agreement. All parties to the consensus can agree in advance (by choosing which software to run) to always honor one possible valid outcome over another possible valid outcome based on a decision rule. From a legal perspective this is more like a judgement of fact from a jury at

[26] Satoshi Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System* (Nov. 2008) p. 2 https://bitcoin.org/bitcoin.pdf.

trial.

For example, Bitcoin's core software does not tell you when any particular valid transaction comes before another valid transaction in the order-keeping ledger of all historical transactions. This order is, nonetheless, critical to determine who paid whom first. Instead of using an automatic rule to settle uncertainties regarding transaction order, Bitcoin's software specifies a decision rule to resolve debates over which valid transaction came first.[27] Specifically, the Bitcoin software calls for a *repeated leader election by proof-of-work*, which we will discuss in a moment while outlining proof-of-work consensus. For now, it's important to simply understand that there are various ways of establishing a decision rule in order to reach consensus over the authoritative state of a decentralized computing system when multiple valid states are possible. All currently employed methods fall into four broad categories: (A) proof-of-work, (B) proof-of-stake, (C) consortium consensus, and (D) social consensus.

### A. Proof-of-Work

As just mentioned, Bitcoin employs a *proof-of-work leader election* as the decision rule for determining the order of valid transactions in the blockchain. Such a consensus method might be useful for various decentralized computing systems, but Bitcoin allows us to describe a working example. *Leader election* means that one participant's record of which transactions came first, second, third, *etc.,* will be selected by all other network participants as the authoritative order of transactions for some designated period of time (beginning with that participant's successful election as leader and ending with the next leader election). We can see how this is a rule of decision, it says essentially: *whenever there is disagreement over two alternative but valid outcomes, defer to the chosen leader's choice for the given period.*

Proof-of-work is the specific method found in the Bitcoin protocol that describes how a leader is periodically chosen.[28] The proof-of-work system is essential to keeping the consensus mechanism *public*. This "election" is, therefore, not anything like the democratic political process to which we are accustomed. After all, if users come and go, freely connecting to the public network without identifying themselves, how would we ever keep track of who is who, or who is trustworthy and deserves our vote? So instead of having a vote, the network holds a lottery where there will be a random drawing and a winner every so often (roughly every 10 minutes for Bitcoin and every 12 seconds for Ethereum).[29]

The term *leader election* is the correct computer science term for this architecture,[30] but for the

---

[27] *Id.* at 2-3.
[28] *See* Nakamoto *supra* note 26 at 3 ("The proof-of-work also solves the problem of determining representation in majority decision making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs").
[29] *See* Vitalik Buterin, "Toward a 12-second Block Time" *Ethereum Blog* (July 2014) https://blog.ethereum.org/2014/07/11/toward-a-12-second-block-time/.
[30] *See* Indranil Gupta, Robbert van Renesse, and Kenneth P. Birman, "A Probabilistically Correct Leader Election Protocol for Large Groups," *Technical Report, Cornell University* (April 2000) ("The classical specification of the leader election problem for a process group states that at the termination of the

rest of us that sounds like something that involves voting and majorities rather than probabilities and lotteries. For clarity we will use the term *leader lottery* from here onwards.

Selecting a periodic leader via lottery in the real world would be easier than finding one on a peer-to-peer network. We could all meet in a room, introduce ourselves, and make it real simple by having everyone put their names in a hat and have one blindfolded person pull out a winner.

That simplicity doesn't work online. If all our peers on the network are putting names in a digital hat, we have no idea if each digital name matches one-to-one with a real person.[31] We could reasonably expect some less-than-scrupulous individuals to make up a bunch of random fake names and stick them in the hat. In the digital world we'd have no way of knowing whether Alice, Beth, Chuck, Dana, and Eve are each real individuals or merely pseudonyms (*i.e.*, "sock puppets") made up by Alice in order have a better chance at winning the lottery. We could try to employ some digital identity system to stop that fraud, but then we would be relying on an external identifier to guarantee the fairness of the system, and that defeats the point of having a public, ungated system to begin with. It would make it costly to participate because you would need to get identified in the real world to do your computing on the decentralized network, and it would force everyone to place trust in the identifier.

Rather than identify all lottery participants and pick names from a hat, we could have a ticket-based lottery, like Powerball. These lotteries only work if the lottery tickets have a cost (if they were free how many tickets to the Powerball would you claim for yourself?). A proof-of-work consensus system merely seeks to make it costly to enter yourself in the lottery. So Alice could still have more than one chance to win, but she incurs real costs every time she buys a new chance.

This has two desirable consequences that help make the lottery a good tool for selecting periodic leaders in a consensus mechanism. (1) *Decentralization:* It would be prohibitively costly to amass enough tickets to ensure that you would be the periodic leader for many repeated periods. (2) *Skin-in-the-game:* Leaders tend to be participants who have made sizable investments in the system by buying costly tickets. Generally speaking, the first reduces the capacity for self-dealing (always putting your transactions first), and the second ensures that the costs of malfeasance are internalized by the participants (who have invested real capital in the long-term success of the platform).

But how do we make those tickets costly when there is no central authority to verify payment? A proof-of-work consensus mechanism imposes costs on participants by making every ticket costly as measured in computing power that provably performs some "work," hence the name proof-of-work. Effectively, every lottery ticket costs one attempt at solving a difficult math

---

protocol, exactly one non-faulty group member is elected as the leader, and every other non-faulty member in the group knows about this choice.").

[31] *See* Nakamoto *supra* note 26 at 3 ("If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs").

problem that can only be solved with guess-and-check.

Think of the Bitcoin lottery ticket as a Sudoku puzzle. To win you need to solve a math puzzle that is difficult (guessing and checking numbers that make rows and columns sum up correctly), but easy for others to check if you have solved it (just sum up the rows and columns). Participants in the network previously agree (with an automatic rule) that the winner of every periodic leader lottery will be the person who first solves the math problem. Ultimately, finding a solution comes down to a lucky guess, but you can make more guesses faster if you have more powerful computers. Because, like Sudoku, it is easy to check someone else's solution, all participants will discover quickly if someone has cracked it, and they will move on to solving the next problem so they can be the leader in the next period.

You might be wondering… *who is setting these problems up?! How is there not an all-powerful algebra teacher controlling Bitcoin?* There isn't, because Bitcoin uses an *open-ended* problem that is specified using only publicly available information found in the Bitcoin protocol software. To extend our classroom metaphor, imagine that the problem on the blackboard is this: *flip a coin heads up 20 times in a row*—a completely open-ended problem. First, we students all agree the problem on the blackboard is the problem we are all competing to solve (an automatic rule), and then once we get flipping, we can all agree if someone does it. Then, once someone "wins," that person is the leader, and we can begin flipping coins again to determine the next leader. We never need a teacher or central authority to present the next problem, we just go ahead and compute the same problem. It's difficult to get less metaphorical or more specific than that without discussing cryptographic functions, something we would like to avoid in this general overview.[32]

What is important to take away from this discussion is that participants enter the lottery by guessing solutions to a publicly posted math problem with their computers, and that more computing power will mean more guesses (more coin flips), which means more chances to win. Because computing power is expensive (both in terms of buying computer hardware, and using electricity to power computing cycles on that hardware) every additional lottery ticket has a cost to the participant.

But if lottery tickets in this leader lottery are costly, then why even participate? After all, the prize for winning would be the right to provide what is effectively a public good: offering an authoritative list of valid transactions on the network for a period of time. This could provide the winner with some benefits (such as ensuring that her own transactions get included in the ledger) but most of the benefits go to the other network participants who get to use a public ledger. So, proof-of-work systems also generally provide a cash reward (in the form of the tokens native to the network) to the holder of a winning ticket, usually called *the mining reward*. This reward can be any fees that were voluntarily appended to transactions by senders on the network (in order to make their transactions more appealing for an elected leader to

[32] For a non-technical but more comprehensive explanation of how the bitcoin proof-of-work process operates, see Peter Van Valkenburgh, "What is Bitcoin Mining, and Why is it Necessary?" *Coin Center* (Dec. 2014) https://coincenter.org/entry/what-is-bitcoin-mining-and-why-is-it-necessary.

include in the section of the ledger she is writing), as well as permission within the software's automatic rules to create new money by sending herself a transaction with no source of funds (socializing the cost of a reward through inflation).[33]

Bitcoin users who decide to participate in this leader lottery have come to be called Miners because they perform "work" in return for newly created value. The label, however, belies the larger role these participants play in generating and maintaining consensus across the decentralized computing system. Both the work and the reward are secondary technical features necessary to the creation of a decentralized mechanism for picking periodic leaders who can ensure that data discrepancies between participants are quickly and fairly resolved.

Without a reward baked into the conesus mechanism, it is hard to understand why users would be incentivized to participate honestly in maintaining the network. Much fuss has been made over developing a "blockchain without the bitcoin," as if the currency aspect of the network pollutes what would otherwise be a useful network technology with an ideology or political agenda (or, at the very, least creates too many regulatory complications to be worth the trouble). But, as we can see, the only way to maintain a public network where leaders need to be periodically selected and rewarded for their participation is to award them with tokens that are native to the network itself (*i.e.*, the transaction history and scarcity of the token are a part of the data over which the consensus network is continually coming to an agreement). If participants are rewarded with assets that exist only according to data structures outside the network (*e.g.*, dollars or yen, the balances and scarcity of which are described in the balance sheets of banks) then we've reintroduced the need for identified parties who must be trusted to perform the rewarding function honestly and without bias.

Public blockchain networks need scarce tokens for technical reasons, not (merely) because their proponents may have political or ideological motivations for supporting alternative currencies. Ethereum, for example, is a public consensus-driven decentralized computing network that aspires to provide several user applications aside from electronic cash (*e.g.*, identity management,[34] reputation accounting,[35] community governance,[36] etc.), but it still has

---

[33] Recall that this is a violation of the automatic rule we discussed earlier in Bitcoin—this is the one exception to that automatic rule, you can send funds without referencing a funding source if and only if you won the leader lottery for the period when you send the transaction; this special transaction is called a coinbase transaction and the amount you are allowed to send is capped according to the monetary policy of the cryptocurrency—yet another automatic rule in the software.

[34] *See*, *e.g.*, Thomson Reuters, *BlockOneID for Ethereum: An identity mapping service for Ethereum blockchains,* https://blockone.thomsonreuters.com/ *last accessed* Dec. 2016.

[35] *See*, *e.g.*, Jack Peterson and Josephf Krug, *Augur: a Decentralized, Open-Source Platform for Prediction Markets*, http://bravenewcoin.com/assets/Whitepapers/Augur-A-Decentralized-Open-Source-Platform-for-Prediction-Markets.pdf.

[36] *See* Vitalik Buterin, "An Introduction to Futarchy" *Ethereum Blog* (Aug. 2014) https://blog.ethereum.org/2014/08/21/introduction-futarchy/ ("Although our modern communications technology is drastically augmenting individuals' naturally limited ability to both interact and gather and process information, the governance processes we have today are still dependent on what may now be seen as centralized crutches and arbitrary distinctions such as 'member', 'employee', 'customer' and

a scarce token that rewards winning participants in the leader lottery: ether. A blockchain without bitcoin or similarly scarce token is a private network, essentially a shared database with pre-identified and authenticated users.

To recap, a public consensus method should allow anyone to participate without obtaining some sort of credential from an external identifier. Without identification, however, a user could pretend to be several users and gain an unfair advantage in the leader lottery used to reach agreement when there are disputes over two or more valid outcomes (like alternative orders of transactions in a ledger). To deal with this problem, participation in the leader lottery is made costly by demanding that participants solve difficult math equations that will require costly hardware and electricity—proof-of-work. As a result, it (A) becomes too expensive to dominate the lottery by obtaining a substantial number of tickets, and (B) ensures that lottery winners are invested in the long-term success of the decentralized computing system. Winning participants are, in turn, rewarded with a scarce token native to the network.

### B. Proof-of-Stake

Now that we have an intuitive understanding of proof-of-work consensus, it is fairly simple to explain the general mechanism behind proof-of-stake consensus. Recall that the goal behind proof-of-work is to make participation in the consensus costly. If the consensus mechanism involves a leader lottery, then we employ proof-of-work to make buying up all the lottery tickets prohibitively expensive.

Proof-of-stake systems are also designed to make participation come at the cost of some provable sacrifice. Instead of requiring calculation in exchange for a lottery ticket, a proof-of-stake mechanism requires that participants prove that they hold and/or can temporarily forgo access to a valuable token that travels on the network.[37] So if Bitcoin was a proof-of-stake-based cryptocurrency, then participation in the lottery could require users to stake some of the bitcoins they control—to prove that they control or to sacrifice their control over those valuable funds. The mechanism could demand that participation requires merely a mathematical proof that the user has possession of these tokens on the blockchain, or it could

---

'investor' – features that were arguably originally necessary because of the inherent difficulties of managing large numbers of people up to this point, but perhaps no longer. Now, it may be possible to create systems that are more fluid and generalized that take advantage of the full power law curve of people's ability and desire to contribute. There are a number of new governance models that try to take advantage of our new tools to improve transparency and efficiency, including liquid democracy and holacracy; the one that I will discuss and dissect today is futarchy.").

[37] *See* Vitalik Buterin, "What proof of stake is and why it matters" *Bitcoin Magazine* (Aug 2013) https://bitcoinmagazine.com/articles/what-proof-of-stake-is-and-why-it-matters-1377531463 ("Rather than requiring the prover to perform a certain amount of computational work, a proof of stake system requires the prover to show ownership of a certain amount of money. The reason why Satoshi could not have done this himself is simple: before 2009, there was no kind of digital property which could securely interact with cryptographic protocols. Paypal and online credit card payments have been around for over ten years, but those systems are centralized, so creating a proof of stake system around them would allow Paypal and credit card providers themselves to cheat it by generating fake transactions.").

demand the permanent relinquishment or even destruction of these token (something often referred to as "proof-of-burn"[38]), or it could be a temporary stake, effectively a bond (*e.g.*, I stake 50 bitcoins—and thereby relinquish my ability to spend them—for the next 150 cycles of the leader lottery at which point I will regain control over the coins and can decide whether to stake again in the future). Regardless of how exactly it is specified, the goal is to use the value of the tokens (rather than the cost of computing) as the provable signal necessary for participation in the leader lottery.

If the tokens that travel on this decentralized network are available for sale on a variety of competitive exchanges (whether in exchange for dollars, euros, or other cryptocurrencies) or can be obtained by free transfer from existing users (whether as a gift or in payment for labor or some valuable good) then anyone with sufficient economic resources can, in theory, join the consensus, because they can obtain the tokens necessary to offer a proof-of-stake. In this sense, proof-of-stake consensus methods are, like proof-of-work methods, public.

### C. Consortium Consensus

Consortium systems have a simpler solution to making lottery-style elections fair: only allow identified parties to participate. If we decide to trust an outside authority to identify all consortium members, provisioning members with cryptographic keys which they can use to sign their communications and prove authenticity, then we can run software that would only grant lottery tickets to participants who send validly signed messages.[39] We know Alice, Beth, Chuck, Dana, and Eve are each real individuals because we previously provisioned them each

---

[38] *See* Counterparty, "Why Proof-of-Burn" *Counterparty Blog* (Mar. 2014) http://counterparty.io/news/why-proof-of-burn/.

[39] When all parties are identified and can be trusted we may not even need a provably fair leader lottery; the leader could simply be the consortium participant with the best quality connection to the network, or it could rotate according to a pre-established order, or an upcoming schedule of leaders could be picked by an offline meeting of participants every year. Indeed, the identified parties could simply choose to use one of the many pre-blockchain fault-tolerant consensus protocols, *e.g.* Paxos, which have a long (around 25 years) and established track record (*see* Pease *supra* note 11), or perhaps simply a basic distributed database tool, *e.g.* an Oracle Database product. It is the longstanding availability of these tools and their persistent non-adoption by the financial industry that has spurred many to cynically characterize the present enthusiasm for permissioned blockchains as nothing more than a bitcoin-inspired and blockchain-branded pitch for selling marginally improved infrastructure to conservative institutions. *See*, *e.g.*, Wences Casares, (Panel Remarks) *Tech Crunch Disrupt: Is it time to stick a fork in Bitcoin?* (Sep. 2015) https://www.youtube.com/watch?v=ORcFGBhDDis ("That's called a private database, and it has existed for a long time. What's new about Bitcoin is that it's a decentralized, trustless ledger. The second you do it your own it's called a private database, and they have existed for a very long time. There's nothing revolutionary about that. … If you're a Visa executive, Bank of America executive, or a Wells Fargo executive, it has become very fashionable to say, 'I really, really like the blockchain. I'm very interested in the blockchain, but I'm not interested in bitcoin,' which is the equivalent of saying, 'I really like the browser, but I don't like the Internet.' It's ridiculous. Those people don't want to be the ones who didn't see the Internet coming, and they want to say something nice about it without saying something nice about it. They don't realize that the blockchain does not work without bitcoin. The blockchain is the first decentralized, trustless database because the miners maintain it, and the miners do so because they get paid in bitcoin. Even though there are a lot of nice use cases on top of that, none of them work without the miners being paid with bitcoin.")

with secret keys and to obtain a lottery ticket each signs a message with his or her unique key.

This consortium method avoids the costs of solving math problems or staking valuable tokens that is inherent in proof-of-work and proof-of-stake systems.[40] The consortium method, however, also reintroduces permission and trust into the decentralized computing system. We need to be identified and granted access to the network in order to participate and we need to trust that the party tasked with making these identifications is acting fairly.

### D. Social Consensus

Finally, we come to the last general category of consensus mechanisms, social consensus. You can think of the social consensus mechanism as somewhere in between the fully identified and permissioned consortium model, and the fully pseudonymous and public proof-of-work and proof-of-stake models.

Like the consortium model, you choose to trust some identified participants rather than relying on pseudonymous participants who offer a costly signal of credibility. Unlike the consortium model, however, each individual is her own identifying authority; she can choose which counterparties she trusts and build a social network of those with whom she feels comfortable entrusting the role of writing new data to the blockchain (or agreeing on some computation generally). We might then expect various users with differing social networks to disagree over the authoritative state of the consensus data, but the network can be designed to come to global agreement by looking for a subset of all transaction or computation data that some minimum number of trusted participants (perhaps a majority or a supermajority of trusted participants on the network) have agreed upon.[41]

As with proof-of-work and proof-of-stake consensus mechanisms, a social consensus mechanism will generally be public. Anyone can join but they must be selected as trustworthy by some minimum number of participants before they can participate in full.

## III. Publicness, Trust, and Privacy Across Various Consensus Models

We've spent a good deal of time outlining these various consensus models because the specifics of their architecture will inevitably have meaningful consequences for the applications that are built on top of them, and, by extension, the people who will use those applications. One does not simply procure some "blockchain technology" to build better digital identity systems, property registries, voting infrastructure, or any of the other ambitious killer apps that have been proposed and widely touted for this technology. Building

---

[40] *See* Sams *supra* note 16.

[41] *See*, *e.g.*, the Ripple Protocol's consensus mechanism. David Schwartz, Noah Youngs, Arthur Britto, *The Ripple Protocol Consensus Algorithm* (2014) https://ripple.com/consensus-whitepaper/ ("Each server, maintains a unique node list (UNL), which is a set of other servers that s queries when determining consensus. Only the votes of the other members of the UNL of s are considered when determining consensus (as opposed to every node on the network). Thus the UNL represents a subset of the network which when taken collectively, is "trusted" by s to not collude in an attempt to defraud the network.").

any of those applications will require either (A) the modification and use of an existing consensus network (*e.g.*, build the application on top of Bitcoin or Ethereum) or (B) the creation of a new consensus network (both the development of consensus software and the bootstrapping of a network of peers who run the software that generates the consensus). The choice of whether to use one of the existing *public* (*i.e.*, proof-of-work, proof-of-stake, or social consensus) networks, to create a new *public* network, or to design and implement a private consensus network will be a choice that affects the relative publicness of the application, the degree of trust that users must place in other users or maintainers of the application or the underlying network, and the degree of privacy that the application is capable of offering its users. Each of these key consensus mechanism attributes, publicness, trust, and privacy will now be discussed in turn.

### A. Publicness Across Consensus Mechanisms

Speaking generally, public consensus-driven decentralized computing systems are exciting and disruptive because their publicness resembles the early Internet. As we described previously, the Internet became the vibrant ecosystem we know today largely because it is so easy to build hardware or software that can seamlessly integrate with TCP/IP, the lower level networking protocol (language) that powers the network. That lower level is pseudonymous. Devices connect to the network and are automatically assigned a seemingly random number rather than a real-world identity.[42] The lower level is permissionless. Devices can send or receive data to and from any other pseudonym so long as the messages conform to the protocol specification.[43] The lower level is general purpose and extensible. TCP/IP only describes how packets of data should move through the network. It does not dictate what the contents of those packets can or should be.[44] Higher level protocols can be built on top of TCP/IP to interpret sent data as web pages, links, videos, emails, SWIFT bank messages,[45] anything that can be imagined, invented, and digitized.

The similarity of TCP/IP to Bitcoin, Ethereum, or any other public blockchain network should be apparent. These systems are also pseudonymous. Users are assigned random but unique cryptographic addresses.[46] These systems are also permissionless. Users can read or write data to the blockchain at will, sending or receiving transactions without seeking the permission of any centralized party. And these systems are also general purpose and extensible. Several parties are building new applications and application layers on top of the Bitcoin network,[47]

---

[42] Crawford *supra* note 25.
[43] W3C *supra* note 21.
[44] *Id.*
[45] Starting in the late 90s several standardized bank messaging services and cooperatives transitioned or adapted their systems to utilize TCP/IP as an underlying networking protocol. SWIFT messages travel over SWIFTNet a higher level Internet protocol that runs on top of TCP/IP. Additionally, the network that supports Fedwire messages, FEDNET, and CHIPS (the international Clearing House Interbank Payment System) network are both built to run on top of TCP/IP. *See* Roy S. Freedman, *Introduction to Financial Technology* (Apr. 2006) pp. 241-246.
[46] Here is an example of a bitcoin address: 1CPwNACt62wts2yGbz1vUuqeGD58SzzeAL.
[47] *See*, *e.g.*, Lerner *supra* note 4, and Ali *supra* note 5.

and Ethereum is explicitly designed to be a flexible foundation for building any trust-minimized application.[48]

In the previous section we classified four types of consensus mechanism into two groups:

- **Public:** Proof-of-work, Proof-of-stake, Social Consensus
- **Private:** Consortium Consensus

Decentralized computing systems built using public consensus mechanisms will, in general, be available to any participants who have an internet-connected device and free software that is compatible with the network. Systems built using a private consensus mechanism will, in general, only be available to participants who have previously identified themselves offline and been granted some form of credential by the identifying authority, which they can use to authenticate their identity whenever they connect to the network.

This characterization of publicness lacks, however, an important nuance. There are basically only two things that any user or potential user might want to do with a decentralized computing network: (1) write data to the network and have it included in the consensus-derived data structure or blockchain, or (2) read data from that network's consensus-derived data structure. Accordingly, a Bitcoin user making a transaction is *writing* new data to the Bitcoin blockchain while a user who queries their balance to confirm payment receipt is *reading* data from the blockchain.

Some have characterized networks where users can freely write consensus data as "permissionless." That is in contrast to "permissioned" networks where users need off-network identification and authentication in order to write. Read access is then characterized as public (anyone can read consensus data) vs. private (only identified and authenticated participants can read consensus data). These terms, however, can be confusing (is a network that has public read-access but private write-access truly public?) so we will continue to use public only in cases where both reading and writing are open to general participation and private in all other cases. For clarity we can summarize this more nuanced characterization with a four-by-four matrix:

[48] *See* Buterin *supra* note 6.

| | | **Writing Data Requires:** | |
|---|---|---|---|
| | | Internet-connected device, free software, and proof-of-work or proof-of-stake. | Off-network Identification, Authentication, and Permission. |
| **Reading Data Requires:** | Internet-connected device and free software. | **Public** (Permissionless, Public Blockchain) | **Public for Reading, Private for Writing** (Permissioned, Public Blockchain) |
| | Off-network Identification, Authentication, and Permission. | **Public for Writing, Private for Reading** (Permissionless, Private Blockchain) | **Private** (Permissioned, Private Blockchain) |

Note an important subtlety in this chart. Public for reading is characterized as requiring only that the reader have an Internet-connected device and free software, while public for writing requires those things but also a proof, either of work or of stake. Bitcoin and Ethereum both exhibit this form of read/write publicness. Anyone with an Internet-connected device and free software can connect to these networks and download the full set of consensus data, *e.g.* the blockchain or list of all valid transactions made on the network from its start. Writing new data to these networks is not quite as easy. If one wants to truly be the node on the network that adds new data to the blockchain, one will have to be selected in the leader elections described in the previous section.[49] So, to truly write new data on these networks one must provide a proof (of computer work or of stake in the network's native token) and then be selected in the network's leader lottery. Even then, however, the user will only truly *write* data to the blockchain for those periods in which she has been chosen as leader.

This, however, is an overly pedantic description of who may write data on these networks. Thousands of people *do* write data to these public blockchain networks without ever running a node that makes a proof, *i.e.* mining. This is because anyone can send a new transaction message to various peers on the network and reasonably expect that the transaction will be picked up by a proof-making node, *i.e.* a miner, who will then incorporate it into a block of transactions which will then be added to the blockchain when that miner wins the leader lottery for a given period. Non-mining peers who want to ensure that their transaction will be written to the blockchain quickly can attach a fee to that transaction which will reward the

[49] *See infra* at 17.

miner who wins the leader lottery and is the first to incorporate the transaction in the blockchain.[50]

Relying on these proof-making nodes to write data may seem like a kind of permissioning, and it is true that any particular user who is chosen in the leader lottery can, for that period, decide which new data will and which new data will not be written to the blockchain. Taking Bitcoin for example, it is true that for the duration that a miner wins the leader lottery, she can censor or block any other user from transacting.

There are two factors that make these systems permissionless in spite of the power of miners or proof makers to block or screen write-access: self-interest among competing proof makers, and ignorance of the data that enters the blockchain.

**Self-interest.** If a user wants to ensure that her transaction will be added to a public blockchain, she can append a fee to the transaction.[51] Miners or proof makers on the network compete with each other for the block rewards that come with winning the leader lottery. Block rewards are comprised of any fees that were appended to transactions as well as any new money being created through programmed inflation. It is with these block rewards that miners can finance the expensive hardware and electricity necessary to perform competitive proof-of-work calculations or justify the costly sacrifice of tokens necessary in making a proof of stake. Blocking transactions will reduce the fee-revenue component of the block reward, leaving censorship-favoring proof makers at a competitive disadvantage. Therefore it goes against the self-interest of proof makers to selectively censor (*i.e.*, permission) the network. Additionally, to the extent that a network is famed for being censorship resistant, *e.g.* Bitcoin,[52] negative publicity from a proof maker's decision to censor transactions may erode faith in the network as a whole. This could cause the market price of the network's tokens to fall, thereby reducing the real value of the proof maker's returns and/or motivating the community to enforce anti-censorship norms by shaming the offending proof maker.

**Ignorance.** Proof makers may not have very much information about the data they are writing to the chain. In other words, the proof maker may know that a particular transaction is valid (because the digital signatures are valid and the sending address is appropriately funded) but she may have no way of knowing who the real-world sender or recipient in the transaction could be. As we will discuss in the section on privacy,[53] new technologies such as zero-knowledge proofs, could ensure that proof-makers as well as the public can gain effectively no information from the blockchain aside from a proof that all transactions are valid according to the consensus rules of the protocol. In this situation, proof-making or mining become an activity divorced from any sort of off-network or personal decision making,

---

[50] *See* Nakamoto *supra* note 25.
[51] *Id.*
[52] *See, e.g.*, Rainey Reitman, "Bitcoin – a Step Toward Censorship-Resistant Digital Currency" *EFF Deeplinks Blog* (Jan. 2011) https://www.eff.org/deeplinks/2011/01/bitcoin-step-toward-censorship-resistant.
[53] *See infra* at 35.

people simply run machines that always add data to the blockchain if it is valid according to the rules of the protocol and are never in a position to discriminate against users for any other reason.

It's simply not necessary to go into this highly nuanced analysis when it comes to consortium-based consensus mechanisms. By definition, these systems will be permissioned at the write-level because only previously identified participants can participate in the consensus. A choice could then be made by the designers of the system, to make read-access to the results of that consensus public or private.

### B. Trust Across Consensus Mechanisms

Early decentralized computing systems, like Bitcoin, are designed for serious uses. These networks custody people's valuables, help them move their money.[54] These networks may soon keep track of their users' identity credentials,[55] and eventually even—in the case of the Internet of Things—help them control their door locks, their baby monitors, their cars, and their homes.[56]

A fundamental design goal of these systems is to decentralize control over the network such that a user will not need to trust a bank-like company's honesty in order to safeguard her money,[57] or trust a technology company in order to safeguard access to her smart home devices.[58] Who or what do you trust to guarantee these systems if not a reputable intermediary, and how does that model of trust change depending on the type of consensus mechanism employed in the system's design? These are the questions addressed in this subsection.

To start, any discussion of trust must deal with three essential subtopics:

- **Software:** Every system described in this testimony is built from software, and the auditability of that software, as well as the nature of the process of writing that software is the first concern we should have when we ask ourselves: can I trust this system?

---

[54] *See infra* at 45. *See also* Nakamoto *supra* note 26.
[55] *See infra* at 51. *See also* Ali *supra* note 5.
[56] *See infra* at 58. *See also* Peter Saint-Andre, "How can blockchains improve the Internet of Things?" *Coin Center* (Oct. 2016) https://coincenter.org/entry/how-can-blockchains-improve-the-internet-of-things.
[57] *See* Nakamoto *supra* note 26 ("What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party.").
[58] *See* IBM Institute for Business Value, *Device Democracy: Saving the future of the Internet of Things*, https://public.dhe.ibm.com/common/ssi/ecm/gb/en/gbe03620usen/GBE03620USEN.PDF ("The Internet was originally built on trust. In the post-Snowden era, it is evident that trust in the Internet is over. The notion of IoT solutions built as centralized systems with trusted partners is now something of a fantasy. Most solutions today provide the ability for centralized authorities, whether governments, manufacturers or service providers to gain unauthorized access to and control devices by collecting and analyzing user data.").

- **Consensus:** The software describes what we have called automatic rules and decision rules. The administration of these rules and the creation of consensus amongst the participants of the system is our second concern with respect to trust.
- **Purpose: "**Trust" or "trustworthiness" is not a monolithic whole. The parties to the system may demand varying requirements from the system: a system to operate an office sports betting pool may not need to be as trustworthy as a system for executing interest-rate swaps among banks. Additionally, the parties to the system may have a good reason to put faith in their fellow participants, and therefore they may not need a system designed to fully supplant trust in one's counterparties.

#### i. Trust in Software

As a first pass, it is important to recall that much of the agreement between participants in these systems is established by what we called automatic rules that are specified in the software. Additionally, we must remember that decision rules will also always be described in the software, even if the decision-making process is then carried out by network participants (whether through proof-of-work, proof-of-stake, consortium, or social consensus means). The software is therefore, to make another legal analogy, the constitutional law of the network; it describes the process by which all subsidiary legal structures should and will ultimately function. The software is always the first element of the system that we must consider when judging the system's relative trustworthiness.

As a general rule, open-source software (*i.e.*, software whose source code can be viewed and audited by any and all interested parties free of any need to seek a copyright license or permission from a patent holder) may be preferable in the context of decentralized systems.[59]

[59] There is a vibrant debate over the relative security of open vs. closed source software in general, and strong arguments on both sides. We take no position in this debate. In the specific context of decentralized networks, however, open source software may have an advantage. In a typical, centralized computer system there will be one entity who, as an individual, business, or institution, is legally accountable to the users of its products and therefore motivated to carefully procure software tools, establish relationships with reputable vendors and/or design software in house, and ultimately audit the tools they chose to implement in their system, whether they be open- or closed-source. In a decentralized system and then agree on which solutions to use. These unaffiliated individuals may not share the same level of trust in a particular vendor of closed-source software. Geographically and culturally diverse, participants may not share the same capabilities for legal recourse against a vendor in the event of negligence, and they may not be able to rely on the vendor for support in the event of a failure that affects them disproportionately to the rest of the network. Popular open-source software projects do not rely on the reputation of a particular vendor to establish trust. Instead, an open community of participants independently develop and audit the code. Open source software is, by definition, publicly available for audit, and would therefore allow the several uncoordinated stakeholders in a decentralized computing system to more easily judge the source code and make decisions for themselves regarding security. Even the developers of *private consensus mechanisms* have felt it prudent to nonetheless make their *software open-source*, likely for this very reason: they need to convince several unaffiliated parties (*e.g.* a consortium of banks) of the software's fairness and validity, while assuaging fears of vendor lock-in. *See*, *e.g.*, Jemima Kelly, "Exclusive: Blockchain platform developed by banks to be open-source" *Reuters* (Oct. 2016) http://www.reuters.com/article/us-banks-blockchain-r3-exclusive-idUSKCN12K17E.

Open-source practices provide an opportunity for developer transparency, an opportunity for a developer or group of developers to put their cards on the table and show with precision what it is they are building. It also subjects that design to an unbounded set of potential security auditors who may detect innocent mistakes as well as malicious backdoors.[60] Without visibility into the software we are putting a good deal of faith in the person selling us that software or advocating for its use. Closed-source software, also referred to as proprietary software, may be superior for various applications (*e.g.*, a word processor or a game), but for decentralized applications that we intend to trust with our money, reputation, identity, or any other valuable agreement between users, close- source software creates real risks. To extend our legal metaphor, a closed-source consensus protocol is not unlike a constitution that no one in the country is allowed to read without seeking permission from the drafter or central government.

To give a real-world example, imagine if someone decided to create an alternative to Bitcoin by copying and modifying the Bitcoin software. What if this person changed the automatic rule that requires all transactions to be funded by prior transactions, to a rule stating that one particular pseudonymous participant would be allowed to send transactions out of thin air. If we are going to use this bizarro-Bitcoin as a shared currency, we would certainly want to know that this change to the software's automatic consensus rules has been made. Our new bizarro-Bitcoin network is now allowing one special user to print money to her heart's content. If we have no way to freely read and audit that code (or to rely on a diverse range of third-party validators to do that audit independent of the software author) then we have no reason to trust the network it creates or the agreements it powers.

#### ii. Trust in the Consensus

After looking at the software, we next need to judge the trustworthiness of the consensus mechanism implemented by the software. Regardless of what some more fervent advocates of these new technologies may say, no system is truly "trustless." No system relies purely on "math" or "cryptography" to ensure that the agreement reached by the network is in any way just or perfect. Instead, these systems are designed to be *trust minimizing*, designed to rely as little as possible on the honesty of the network's participants, usually by making deceptive or fraudulent participation go against the economic interests of the participants. So, aside from being public or private, we can also discuss how each category of consensus mechanism attempts to minimize trust.

In proof-of-work and proof-of-stake systems, so long as we believe that the participants who together control a simple majority of the total computational power on the network (for proof-of-work) or the staked token value on the network (for proof-of-stake) are behaving honestly, then the network's decision rules will work as intended. The need for trust in the

[60] The idea of security by way of massive public auditing and transparency has come to be called "Linus'Law" and it is commonly expressed as "Many Eyes Make All Bugs Shallow." See Jeff Jones, "Linus's Law aka 'Many Eyes Make All Bugs Shallow'" Microsoft Cyber Trust Blog (Jun. 2006) https://blogs.microsoft.com/cybertrust/2006/06/07/linuss-law-aka-many-eyes-make-all-bugs-shallow/.

network's participants is obviated so long as half of its participants are not united in trying to attack it. If a dishonest party or parties assumes control of a simple majority of the computational power or staking ability on the network, then they can effectively control the outcome of all decision rules, and the results may differ substantially from the expectations of honest participants.

To take Bitcoin as an example, a party with majority control of the network's total computational power could: (1) refuse to put certain transactions into the shared ledger indefinitely, (2) consistently favor her own transactions over others in the speed with which they are recorded in the ledger, and (3) periodically rearrange the ledger's order going back as far in history as she has had the majority of power on the network.[61] She cannot, however, violate the automatic rules on the network: she cannot spend other people's bitcoins, nor can she create more bitcoins than would normally be allowed under the monetary policy rules of the software. By sending messages that violate these automatic rules, she loses compatibility with the network and ceases to take part in the consensus mechanism that enforces decision rules like transaction order.

So in proof-of-work and proof-of-stake systems, we can generally trust that the shared computation is valid and fair so long as we believe it is cost-prohibitive for a malicious actor to amass sufficient computing power or staked tokens to have a majority on the network.

Proof-of-stake systems still lack a robust working prototype. The most notable system, Peercoin, suffered a spate of attacks and reverted to a state where the developers created a whitelist of permissible stakers (effectively a consortium model).[62] Some theorize that a robust proof-of-stake consensus mechanism is an impossible goal, but considering that is beyond the scope of this testimony.[63]

The availability of what is called "forking"[64] adds an additional wrinkle to the question of trust

---

[61] This is commonly referred to as a 51% attack. The limited ability to do harm and exorbitant cost of the attack, combined with the ease with which an attack would be noticed by the community and resolved with modifications to core software lead many to believe that such attacks should be low on the list of threats to the security and trustworthiness of the Bitcoin network. *See* Gavin Andresen, "Neutralizing a 51% Attack" *GavinTech* (May 2012) http://gavintech.blogspot.com/2012/05/neutralizing-51-attack.html; *see also* Daniel Cawrey, "Are 51% Attacks a Real Threat to Bitcoin?" *Coindesk* (June 2014) http://www.coindesk.com/51-attacks-real-threat-bitcoin/.

[62] Andrew Poelstra, "A Treatise on Altcoins" 14 (Mar. 2015) https://download.wpsoftware.net/bitcoin/alts.pdf.

[63] For a technical analysis of proof-of-stake systems see Poelstra *supra* note 61 at 14.

[64] This use of "fork" comes from the larger world of free and public source software development, particularly the communities developing Linux, the open source and oft-forked operating system that powers many enterprise computing systems. Forking refers to a decision amongst some developers within an open source project to duplicate the code of that project and maintain it separately in order to create some derivative invention. See Benjamin Mako Hill, "To Fork or Not To Fork: Lessons From Ubuntu and Debian" (May 2005) https://mako.cc/writing/to_fork_or_not_to_fork.html ("The act of taking the code for a free software project and bifurcating it to create a new project is called "forking." There have been a number of famous forks in free software history. One of the most famous was the schism that led to the parallel development of two versions of the Emacs text editor: GNU Emacs and XEmacs.

in networks that utilize public consensus mechanisms. If two or more factions of users on the network fail to reach an agreement over what we have called "automatic rules," then the network will divide in two or more parts. They will share a computational history up until this impasse but, from the time that one faction chooses to alter their software's automatic rules onward, they will forge new and distinct futures. This has been the case in several so-called *hard forks* of cryptocurrency networks.[65]

To understand the trust implications of hard forks, we need an example. According to an automatic rule in the Bitcoin consensus mechanism, which we'll call the *supply rule*, there can only ever be 21 million bitcoins.[66] This hard limit in the code forms the basis of Bitcoin's value proposition: you are willing to hold and trade these otherwise made-up tokens for real goods because their supply is known to be finite. With supply fixed, any demand from a community of users will result in a positive price. If we choose to trust Bitcoin's long-term valuation, we'll have to worry about fluctuations in demand affecting the price, but at least we won't need to worry about an increase in supply diluting the value of our holdings with inflation. The effect of the *supply rule* is to Bitcoin's value as the effect of the earth is to the value of gold when it resists gold-mining.

While it has never happened, we could imagine a fork of Bitcoin where part of the network wants to increase the total supply of bitcoins from 21 to 42 million by changing that automatic rule. We'll call the more-bitcoins partisans KeynesCoiners, and the rest of the users we'll call MiltonBitters. As soon as the KeynesCoiners update their software to incorporate a change in the supply rule, transactions and blocks from a KeynesCoin computer are invalid when received by a MiltonBit machine and vice versa. Both sides of the network recognize a common history of bitcoin transactions, but going forward they will have irreconcilable futures. If you

---

This schism persists to this day.").

[65] The most notorious fork in recent crypto-times is probably the hard fork of Ethereum during the DAO hack in the summer of 2016. In response to a bug in a widely funded smart contract (the DAO), developers offered a change to the core protocol that would effectively unwind the result of that contract on the blockchain and make DAO investors whole. A minority of network participants disagreed with this policy and refused to update their software. The result was a fork of the network and the creation of Ethereum Classic (effectively an alternative version of Ethereum). While the drama generated a good deal of press from those critical of Ethereum or simply interested in these networks, it should be noted that the price of Ethereum two months before (April 18th: $8.44) and two months after the fork (August 18th: $11.06) shows little evidence for an erosion of trust in the network. For more on the Ethereum fork see Joon Ian Wong and Ian Kar, "Everything you need to know about the ethereum hard fork" *Quartz* (July 2016) http://qz.com/730004/everything-you-need-to-know-about-the-ethereum-hard-fork/.

[66] There is no line of code in the Bitcoin reference client that specifically says, "there will only ever be 21 Million bitcoins." Instead, there is language that describes the permissible size of the reward of new bitcoins that miners who mine new blocks can claim in a coinbase transaction. This reward is referred to as a "block subsidy" and it is coded to start at 50 bitcoins per block and decrease by half on a schedule that would result in a final total supply of roughly 21 million total bitcoins at some point in the year 2140. See Bitcoin Core,"main.cpp," https://github.com/bitcoin/bitcoin/blob/master/src/main.cpp, lines 1380-1391 ("Subsidy is cut in half every 210,000 blocks which will occur approximately every 4 years."). See also "Controlled supply," Bitcoin Wiki, https://en.bitcoin.it/wiki/Controlled_supply (last accesed Dec. 2015).

held bitcoins before the fork, you now have bitcoin balances on both networks (because they share a common history before the fork), and you can run KeynesCoin software on one computer while running MiltonBit on another in order to move your bitcoins on either or both sides of the newly forked network.

Does this violate the trust that users placed in the supposedly sacred 21 million limit? It's hard to say. The MiltonBit network remains a working cryptocurrency for users who want to stick with the 21 million limit, and pro-inflation revolutionaries can switch to the KeynesCoin chain. In fact, now users who are indifferent as to a choice between 21 and 42 can choose to wait it out, or to use both, because their bitcoin holdings are in the history of both sides of the fork and will remain on each chain unless they decide to transact using the compatible software of that chain. To use a term from political science, forking facilitates political *exit* rather than *voice*, leaving a community with whom you disagree rather than lobbying for a change to that community's rules.

It's not all rosy, however. When our hypothetical network split in two, the supply curve changed for only one-half of the network but the demand curve for each coin will probably change for both. Some users will want KeynesCoins and dump their MiltonBit holdings on exchange platforms or over-the-counter trades and vice versa. If a sizable chunk of bitcoiners choose team Keynes, then the price of MiltonBits might fall drastically. If the price of the tokens on open exchanges crumbles, so too could the mining power that safeguards the network against attack.

Rational miners will only spend electricity and capital up to the marginal revenue obtained from mining. If the price of the coin with respect to the cost of electricity and hardware declines, miners will probably take their mining machines offline, or if possible, dedicate their efforts to other more lucrative proof-of-work driven cryptocurrencies. If the total mining power on the network is low enough, a bad actor could corner the mining market more easily and attempt to disrupt the consensus system: block transactions at will, reverse transactions throughout the period in which they have control of the majority mining power, etc.

To round up this forking discussion, we can make the following general observation about trust in public consensus-driven networks. These systems do not create absolute trust or absolutely true computation; they merely generate a single source of truth that is trustworthy (A) only amongst participants who choose to remain compatible with their fellow participants and (B) only so long as a majority of those participants are behaving honestly. These systems do not fully obviate the need for "trust," but instead minimize the amount of trust necessary to a presumption that others will continue to run the software you also want to run, and no party will gain sufficient computational resources or stake sufficient wealth to dominate and then manipulate a leader lottery or other decision rules described by that software.

Consortium systems may be similar in that generally they are only trustworthy so long as a majority of identified consortium members are behaving honestly, and will only function if all members continue to run compatible software. However, we must also consider the entity that

identifies and then grants credentials to the consortium members. If this identifying member is corrupted, it could potentially shift the balance of power by granting more participatory rights to one or another consortium member than was assumed to be fair and agreed upon by the other members. The sanctity of a lottery or any other decision rule is only upheld by trust in an identifying agent and the safekeeping of identity credentials by participants (rather than by provable sacrifice of resources by participants). As the developers of Monax, a permissioned blockchain platform, explain:

> The security model for permissioned blockchain networks is very similar [to public consensus networks], namely it is the non-predictive distribution of power over block creation among nodes unlikely to collude. Only, in a permissioned blockchain network the barrier to entry, and/or barrier to control, are provided either out of band by a previous or emergent agreement; added to the genesis block of the blockchain network and/or updated over time as different evolutions of the network become necessary. A possible attack vector at this point for overtaking a permissioned blockchain is thieving (or brute forcing) of 2/3rds of the private keys for the validator set."[67]

Additionally, the nature of an identified consortium may make it easier for some subset of the consensus members to find each other and collude to defraud the rest of the network (at least as compared with a network composed of pseudonymous participants with little or no information about their counterparties).

Finally, social consensus mechanisms are also trust-minimized but in a different manner than the other mechanisms. In a social consensus, you must trust some parties on the network, but need not trust all parties. To the extent that a global consensus is composed of some subset of data that the majority of all trusted participants have validated, we may worry that all participants are blindly placing trust in the same parties without careful consideration of how they should choose. If so, these trusted parties may be able to take advantage of this non-discriminating trust from the network at large and collude to defraud the network just as a majority group could do the same in the other mechanisms we've discussed.[68]

#### iii. Trust for What Purpose?

To round up our discussion of trust, we also need to consider the question: *trust for what purpose?* Decentralized computing systems are potentially (and in some cases already are) useful for a variety of applications: peer-to-peer electronic cash,[69] identity,[70]

---

[67] Monax, *What is a Permissioned Blockchain Network?* https://monax.io/explainers/permissioned_blockchains/ *last accessed* Dec. 2016.
[68] Within the Ripple protocol this issue is, in theory, tempered because trusted validators will have reputations to uphold, and should any validator prove untrustworthy users will simply select alternative validators to place on their unique node list. Ripple Wiki: Consensus https://wiki.ripple.com/Consensus *last accessed* Dec 2016.
[69] *See infra* at 45.
[70] *See infra* at 51.

machine-to-machine payments in the Internet of Things,[71] recording property rights,[72] settlement of stock trades,[73] the settlement of accounts between large financial institutions,[74] and more.

In some applications where all participants are part of a tight-knit community with a limited goal (like settling accounts between banks for example), placing trust in an identified consortium and the party doing that identification may be entirely reasonable. Indeed, it may even be reasonable for the software that generates the consensus to be closed source as long as the identified participants (if not the larger public) feel satisfied that sufficient and independent audits of that code have been carried out to ensure that it does in fact do what its developers and vendors claim.

For other applications, however, trust in a central party may be sub-optimal. It could afford certain parties more power over our lives than we would ideally want. Public consensus models are by no means trustless, but they do decentralize power amongst a larger and open set of parties meaning that we are less likely to find ourselves (our transactions, our data, whatever we compute on the network) at the mercy of a single powerful institution that could either maliciously defraud us or negligently fail to maintain a secure network. There are three particular use cases of blockchains for which the trust-minimization inherent in a public consensus mechanism may prove critical: electronic cash, identity systems, and the internet of things. We discuss these in the final section. First, however, we need to discuss privacy.

### C. Privacy Across Consensus Mechanisms

As we'll discuss in the final section, decentralized computing platforms may come to be the systems we use to safeguard our money, our identity, and our homes. Our daily activities, our credentials, and our transactions represent a wealth of personal data. The choice of consensus model can have repercussions with respect to our privacy. Who will be able to see your transactions if you use Bitcoin? Who will be able to see your comings and goings if you use a smart lock powered by Ethereum? Before we jump into the technical specifics, however, it's important to carefully describe what we mean by privacy, and what sort of privacy protection we would reasonably want or expect from decentralized computing systems.

#### i. Privacy and Context

Privacy is never absolute. Even a hermit who never speaks to anyone cannot avoid being seen

---

[71] *See infra* at 58.
[72] *See* Laura Shin, "Republic Of Georgia To Pilot Land Titling On Blockchain With Economist Hernando De Soto, BitFury" *Forbes* (Apr. 2016) http://www.forbes.com/sites/laurashin/2016/04/21/republic-of-georgia-to-pilot-land-titling-on-blockchain-with-economist-hernando-de-soto-bitfury/#e5b6b4265500.
[73] *See* John Detrixhe, "Scotland to Start Own Stock Exchange Using Blockchain Technology" *BloombergTechnology* (Oct. 2016) https://www.bloomberg.com/news/articles/2016-10-27/scotland-to-start-own-stock-exchange-using-blockchain-technology.
[74] *See* Gendal Brown *supra* note 7.

and scrutinized as she goes about her fishing, foraging or any of the other activities necessary to her survival. So rather than thinking about privacy as the mere ability to avoid public exposure or to keep secrets, let's think of it as the ability to control information about ourselves and our activities. This more nuanced concept is best described by Helen Nissenbaum's term *contextual integrity*.[75] Contextual integrity refers to the ability of an individual to control what information is released and what information is kept private depending on the context of a given social interaction.

Compare, for example, the information we'd want released to our dentist in advance of an appointment with the information we'd want released to our spouse in advance of a night out. These interactions have different contexts: medical and commercial vs. romantic and personal. Therefore, we cannot equate privacy with mere data security. Security simply means withholding some secret. Privacy means controlling to whom and in which situations we choose to reveal those secrets.

Whenever I interact with a decentralized system, I generate information that could become public. If the system is to protect my privacy, then ideally it would only share evidence of my interactions with the minimum set of participants necessary to accomplish my goals and expectations in interacting with the system. It should only share information that is relevant and appropriate within the context of the system as the user understands it.

An example makes this clearer: Let's imagine a system for transferring money. Alice gives money to Bob. Who needs to know what about this transaction? Of course, Alice and Bob need to know the amounts involved and who gets what. Bob also needs to know that the money Alice gave him is real and not a forgery, and he also needs to know that Alice truly gave up that money rather than retaining the ability to spend it. Finally, *everyone* who uses this particular sort of money needs to know that in this transaction no new money appeared unexpectedly, because if Alice somehow managed to both send the money as well as keep it for herself, then the supply of all money has grown and *everyone's* money will be worth a little less because of inflation.

Cash solves these problems by allowing the transaction to occur face-to-face between Alice and Bob. Bob can see that Alice has handed him a ten-dollar note. Bob knows he can walk away with the money and Alice won't be able to get it back. If they perform this ritual behind closed doors, no one else learns about the transaction. Cash notes are designed to make counterfeiting difficult, allowing *everyone* to know with some degree of certainty that no new money was created when Alice and Bob transacted.

Cash doesn't work online because a digital image of a ten-dollar note can be endlessly copied at effectively zero cost. Various solutions for moving money electronically have been developed but, of course, they vary in their ability to respect the privacy of the parties as

[75] Nissenbaum, Helen. "Privacy as contextual integrity." Wash. L. Rev. 79 (2004): 119. *Available at:* http://www.kentlaw.edu/faculty/rwarner/classes/internetlaw/2011/materials/nissenbaum_norms.pdf.

compared with cash.

Alice and Bob can use a bank or several banks in order to account for an electronic movement of money between them. Now Alice and Bob know what they need to know, but the bank also knows about the transaction. If the bank is hacked, the records of the transaction may become public knowledge. Despite having relatively little information to go on, *everyone* must be satisfied that the banks are keeping good records and that they are faithfully serving their role as lenders to maintain the relative scarcity and therefore price of the currency.

Bitcoin is a public consensus-driven peer-to-peer network that creates electronic cash for remote transactions without intermediaries like banks. Bitcoin provides Alice and Bob with the transactional information they need because they can (A) generate and agree on pseudonyms for each other, (B) view a global shared ledger that lists bitcoin balances for all pseudonyms, and (C) only spend balances on that ledger if they have a cryptographic key that matches the pseudonym. Bob knows that Alice has given up the funds because they've moved on the ledger to a pseudonym that only he controls. *Everyone* knows that no new money was created because they can see the transaction moved balances between two pseudonyms but did not create any new bitcoins. *Everyone* could also know the specifics of Alice's or Bob's transactions if the pseudonym(s) used by Alice or Bob can be linked to their name publicly.

Thus we see how three different system architectures (cash, electronic banking, and Bitcoin) all afford the relevant parties to the transaction varying levels of access to and control over the information created by, and necessary for, transacting.

#### ii. Privacy versus Transparency in Consensus

As we defined it, consensus is an agreement over (1) some set of data, (2) modifications to or computations with that data, and (3) the rules that govern that data storage and computation. An essential feature of these systems is that much of the activities of the participants will be fully transparent and verifiable to all participants in the consensus: the history of the data over which we are forming consensus is auditable and my modifications and computations with that shared data will be transparent so that my actions can be verified. It would be impossible for a network to ensure that the agreed upon rules for data storage and computation are being honored without some level of transparency.

To use Bitcoin as an example, if the full history of bitcoin transactions between users is not transparent, then I have no way of knowing whether a specific user purporting to send me five bitcoins has ever, herself, received or mined those five bitcoins. Similarly, if the transaction from this user to me is not incorporated in the ledger, no future recipient of the funds I've just been sent can be assured that I'm good for the money.

Bitcoin is able to have this level of transparency but still offer some privacy to its users because all of the entities transacting or mining bitcoin on the network are represented by pseudonyms. Specifically, to use Bitcoin I will have my Bitcoin software generate one or more public-private keypairs. The private key is the secret I need to have in order to sign for valid

transactions, and the public key is the address or account to which people can send me bitcoins. The public key is a pseudonym. My name may be Peter, but when I transact on the network other machines and users will recognize and address me only by a random string of text:

17kdugRB1fdvqFC1BHkBwjZWm2wbt982AH

The problem with this approach is that if anyone learns that I'm the real person behind 17kdug… then they can look up my full transaction history with that address. One solution has been to use several addresses and never reuse an old address. So everytime I ask to be paid, my Bitcoin software will create a new address for me to share with the payor,[76] and everytime I send bitcoin from an address, the remainder or "change" from the transaction is sent to a brand new address. Even with these procedures in place, however, my several addresses could still be linked and identified with forensic tools. For example, if I have two bitcoins each in three different addresses, and I want to pay someone five bitcoins, I will need to use all three of my addresses in order to fund the transaction. With all three of these addresses listed as inputs to the transaction, a nosey person looking at the blockchain can easily assume with some certainty that those three addresses were all one person, me. If any of those addresses have been previously marked as belonging to me, then we're back at the initial problem: my full transaction history is potentially public information.

The same privacy problem is generalizable to any sort of decentralized computing platform powered by the consensus mechanisms we have so far discussed. The need for transparency and verifiability may conflict with our desire for privacy as we use these systems. As we'll see there are two general approaches to resolving or ameliorating this conflict: *perimeter security* and a variety of new techniques, which we can call *data minimization*.

#### iii. Perimeter Security versus Data Minimization and Selective Disclosure

Faced with an essential trade-off wherein verifiability requires transparency but privacy requires that user-data remain opaque, there are essentially two design options:

1. **Perimeter Security:** Leave all data relevant to the consensus transparent but restrict the set of parties who verify that data to a local and private group of verifiers with whom you are comfortable sharing otherwise private data.
2. **Data Minimization:** Develop tools to only reveal data essential to group consensus if it is absolutely necessary to verification and allow the group of verifiers to be open and global.

Perimeter security follows an older approach in network security generally: *if there are things to be kept secret, we build a secure perimeter, restrict the flow of sensitive information to within*

[76] This is not as inconvenient as it may seem. The wallet software that I use should keep track of all of these addresses and keep the associated private keys secured in a single file (if I'm securing my own bitcoin) or else a company can keep track of this data on my behalf. Either way, when I transact I don't need to worry about a number of addresses and keys, I just spend bitcoins from my wallet.

*that perimeter, only allow authorized parties into that perimeter, and carefully monitor for and prevent breaches*.[77]

Data minimization takes an alternative approach: *we will not rely on a secure perimeter, all information in the system can be presumed to be global and available, but the only information ever put into to the system is the minimum amount of information necessary to accomplish the goal.*[78]

Again, an example will make this distinction clearer. Alice wants to send money to Bob, but wants privacy. A money transmission system with perimeter security would look rather like existing mobile payment applications like PayPal or Venmo. Alice and Bob share the full private details of their transactions with a single verifier, *e.g.* PayPal. PayPal allows Bob to know that Alice has a sufficient balance to send the money, ensures non-repudiation, and by balancing its books gives the public the assurance that no new money was created out of thin air (it was only transferred). As long as PayPal maintains a secure perimeter, the details of these transactions remain private. The downside of this solution is two-fold: (1) we now cannot rely on the larger public to verify the details of the transaction, we must trust the party or group that is within the perimeter (*e.g.*, Paypal), and (2) if the perimeter is ever breached, then all of this data could become public.

A money transmission system employing data minimization instead of a secure perimeter model would look rather like an improved version of Bitcoin. Recall that within Bitcoin, all details of the transactions are public but they are pseudonymous. We have previously discussed how this pseudonymity can be weak and result in the public revelation of an individual user's full transaction history. A system like Bitcoin with more robust data minimization would limit the public data to information that is relevant to consensus and allow the users to choose what additional information they would like to reveal about their specific transaction. Here's what that could look like:

> ***Information Alice needs to know:*** An address where she can pay Bob, confirmation that Bob got paid (in case he tries to claim he didn't).
>
> ***Information Alice does not need to know:*** the balance of Bob's address(es) before or after the transfer.
>
> ***Information Bob needs to know:*** That he's been paid, and that the payment is genuine (the

[77] *See* Lenny Zeltser, Karen Kent, *et al.* "Perimeter Security Fundamentals" *Inside Network Perimeter Security* (Apr. 2005) *chapter available at* http://www.informit.com/articles/article.aspx?p=376256.
[78] *See generally* Peter Schaar, "Privacy by Design" 3 *Identity in the Information Society* 2 (Aug. 2010) *available at* http://link.springer.com/article/10.1007/s12394-010-0055-x/fulltext.html discussing the concept of data minimization within the context of Privacy by Design, *i.e.* "The idea of incorporating technological data protection" into the overall design of an application or computer system, "instead of having to come up with laborious and time-consuming 'patches' later on. … Privacy by Design goes beyond maintaining security. Privacy by Design includes the idea that systems should be designed and constructed in a way to avoid or minimize the amount of personal data processed. Key elements of data minimization are the separation of personal identifiers and content data, the use of pseudonyms and the anonymization or deletion of personal data as early as possible."

sender has enough money to fund the transaction).

***Information Bob does not need to know:*** the name of the sender, the balance of the sender's address(es) before of after the transfer.

***Information the whole network (the public) needs to know:*** That money was transferred but was not created.

***Information the whole network does not need to know:*** Any identities (including pseudonyms) involved in the transfer, or the specific amounts that were involved in the transfer (because these can potentially also be used to identify the transaction).

From this baseline of privacy, the parties should also be able to *voluntarily* choose to be less private. This choice is referred to as *selective disclosure*.[79]

***Alice should be able to choose what otherwise private information she'd like to selectively disclose***:

- She can choose to let Bob know the payment was from her and should be able to prove to Bob (using the verification power of the entire network that she is the one who paid him).
- She can choose to let particular third parties (or the public at large) know the details of the transaction (her name, Bob's name, and/or the amount that was paid).

***Bob should be able to choose what otherwise private information he'd like to selectively disclose:***

- He can choose to let third parties know the details or the transaction (his name, the amount he was paid, and—if Alice shared this information with him—Alice's name).

Similarly, Bob should be able to reject payments if he'd like, this way Bob can refuse to accept a payment from someone who did not identify herself to him. While these disclosures are voluntary as far as the software is concerned, they may be required by law.[80]

This same selective disclosure paradigm could be highly useful in other consensus-driven systems aside from value-transfer, for example identity: a customer should be able to present

---

[79] *See* Zooko Wilcox and Paige Peterson, "The Encrypted Memo Field" *Zcash Blog* (Dec 2016) https://z.cash/blog/encrypted-memo-field.html.

[80] *See*, *e.g.*, Zooko Wilcox and Peter Van Valkenburgh, "What is Zcash" *Coin Center* (Dec 2016) https://coincenter.org/entry/what-is-zcash ("whenever the law demands transparency and whenever proper legal process is followed to obtain that transparency, a user or regulated firm can easily oblige by sharing the view key that un-blinds private transactions with the proper authorities. This is, in many ways, superior to the current state of affairs with Bitcoin where both law enforcement and the general public can see a wealth of private information about your Bitcoin addresses. It's also better than the current state of affairs with pre-blockchain banking transactions because the data being shared can be verified by an open network of computers, rather than law enforcement needing to take the regulated party or the individual being questioned at their word.").

a bartender with an attestation token that proves that an attestor (*e.g.*, the Department of Motor Vehicles) has verified that she's old enough to legally drink, but that token and the decentralized computing system that powers it should not inadvertently disclose her name, address, or anything else about her to the bartender unless she wants to reveal that information.[81]

This architecture has significant advantages over perimeter security. Unlike perimeter security, the choice of remaining private does not come at the cost of trusting a party or a group within a secure perimeter. The validity of the transfer, the fact that no new money was created, and that the transfer cannot be reversed, can all be public information guaranteed by an open set of validators rather than be facts we need to trust a private set of validators to be honest about. Also, with data minimization and selective disclosure there is no central perimeter to be hacked. It's possible that the credentials I use to choose my level of selective disclosure could one day be hacked, and the hacker could reveal all of my transaction records, but there is no central perimeter that, if hacked, would reveal *all private transactions from all users* of the system. The negligence of one user, employee, or vendor partner (failure to set a strong password, willingness to open strange attachments in phishing emails, etc.) does not automatically jeopardize the entire system.[82]

#### iv. Perimeters or Minimization Techniques in Consensus Mechanism Design

It has been suggested that public consensus mechanisms (*i.e.*, proof-of-work, proof-of-stake, and social consensus) are not suitable for enterprise or financial sector applications because they are not sufficiently private.[83] It is true that Bitcoin presents us with an example of this weakness: pseudonyms are too easily identified and transaction histories of users are too vulnerable to public scrutiny. However, faced with this dilemma, there are a variety of solutions. The commonly cited solution is to build only private, consortium consensus-driven

---

[81] David Birch has worked diligently to articulate this notion of data minimization and transactional identity. As Birch frames it: "What is needed to enable transactions is not identity per se but the associated entitlements." Not, "I am John Doe" but instead "I am old enough to order this beer." Birch calls this form of identification
"pseudonyms with credentials." David Birch, Identity is the New Money (2014).

[82] Take for example the 2015 Target breach. At Target, consumer credit card credentials were stored on an internal server, but hackers did not initially infiltrate this server. Instead, they targeted a vulnerable server controlled by a heating and cooling company that Target used as a facilities services vendor. By granting some network access to this vendor, Target unknowingly and unintentionally extended the network of trust to which its customers belonged. Once the heating and cooling company was compromised, so was Target and so were all of their customers. With enough new and variable links in a chain, one is likely to be weak enough to unravel the whole. *See* Brian Krebs, "Target Hackers Broke in Via HVAC Company," *KrebsonSecurity* (Feb. 2015)
http://krebsonsecurity.com/2014/02/target-hackers-broke-in-via-hvac-company/.

[83] *See*, *e.g.*, ESMA, Discussion Paper: The Distributed Ledger Technology Applied to Securities Markets (Feb. 6, 2016) https://www.esma.europa.eu/sites/default/files/library/2016-773_dp_dlt.pdf ("We understand that the DLT [distributed ledger technology] that is likely to be applied to securities markets would be 'permission-based' in contrast to the 'permissionless' system that was originally designed for virtual currencies, *e.g.*, Bitcoins, for a number of reasons, including efficiency, security and privacy purposes.")

networks for these use-cases. The only privacy gain inherent to this approach is the creation of perimeter security. For example, the banking technology consortium R3 has described its Corda decentralized ledger product as follows:

> "The foundational object in our concept is a state object, which is a digital document which records the existence, content and current state of an agreement between two or more parties. It is intended to be shared only with those who have a legitimate reason to see it."[84]

Privacy is thus ensured by sharing the "state object" only with one's trusted counterparties, with those "who have a legitimate reason to see it." The agreement is made private by placing it behind a secure perimeter, not necessarily by limiting the contents of the agreement to data relevant to consensus over that agreement. If any of the "legitimate" parties are compromised, the contents of the agreement could become public. In this sense the consortium model on its own does little to change the state of information security beyond what we see from existing centralized financial intermediaries. Indeed, it may be on balance a more vulnerable system because the secure perimeter now includes employees at other firms. Additionally, if the entire contents of the agreement are private to the relevant parties, independent validation of the data cannot occur in a fully trust-minimized manner (*i.e.*, from an open and global network of impartial transaction validators); one only gets validation from the set of parties permitted by the consortium to enter the secure perimeter.

To R3's credit, it is investigating various other approaches to better enhance privacy as described in their near- to mid-term roadmap:

> Privacy enhancements using technology such as address randomization, zero-knowledge proofs.[85]

These are approaches that apply equally well in consortium and public consensus-driven systems. Significantly, these technologies have been primarily pioneered in the Bitcoin and related cryptocurrency communities.

Address randomization is effectively the attempt to create more robust pseudonyms that fail to yield to forensic identification techniques. Most research into the development of these techniques is occurring in the Bitcoin space where, without robust address randomization, privacy is fairly poor as previously described. Notable pioneering advances in this approach are the CoinJoin[86] and Coin Shuffle[87] protocols, which create decentralized communications

---

[84] Corda Introductory Whitepaper (Aug. 24, 2016) http://static1.squarespace.com/static/55f73743e4b051cfcc0b02cf/t/57bda2fdebbd1acc9c0309b2/1472045822585/corda-introductory-whitepaper-final.pdf.

[85] *Id.*

[86] Blockchain.info, SharedCoin and other CoinJoin implementations: Uses and Limitations (June 10, 2014) https://blog.blockchain.com/2014/06/10/sharedcoin-and-other-coinjoin-implementations-uses-and-limitations/.

[87] Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate, CoinShuffle: Practical Decentralized

channels to facilitate the shuffling of bitcoins between several addresses in a manner that makes it difficult to link a set of addresses to one particular user. Additionally, changes to the Bitcoin core protocol have been researched and proposed that would obscure the value of each transaction as it appears in the blockchain, a project referred to as Confidential Transactions.[88] Simultaneously, some security researchers have proposed that key concepts from the Confidential Transactions and CoinJoin protocols, could be combined and used to obscure both the value and the participants to a transaction. This new research has been referred to, whimsically, as Mimblewimble (from the Harry Potter books) and it is now being developed into a standalone cryptocurrency called Grin.[89]

Separately, Zero-knowledge proofs are a cryptographic tool for proving some important fact (*e.g.*, this transaction is valid, these bitcoins have never been spent by this sender before), without revealing any other information aside from the proof.[90] Integrating zero-knowledge proofs into a public consensus blockchain could potentially allow a decentralized open set of transaction validators to prove that all recent transactions have been appropriately funded, signed, and not double-spent, without revealing any additional information about who sent how much to whom. The Zcash Electronic Coin Company has been pioneering these technologies in the form of Zcash, a public consensus (proof-of-work) driven digital currency network. Not only is Zcash testing the viability of a truly data-minimized approach to privacy and consensus, the protocol also allows users to selectively disclose information about their transactions to whomever they choose.

> Zcash transactions automatically hide the sender, recipient and value of all transactions on the blockchain. Only those with the correct view key can see the contents. Users have complete control and can opt-in to provide others with their view key at their discretion.[91]

Still another cryptographic tool that can be utilized to provide privacy alongside reliable verification of data on a public blockchain is a ring signature. Briefly, these signature schemes can be employed to prove that one of several members of a group authoritatively signed a message without revealing which member of the group actually did the signing. Ring signatures are already employed by the cryptocurrency Monero to protect user privacy.[92]

These systems are in many ways be ideal: Trust in the scarcity of the underlying tokens and the non-reputability of transactions is generated by an open set of impartial validators (rather

---

Coin Mixing for Bitcoin https://crypsys.mmci.uni-saarland.de/projects/CoinShuffle/coinshuffle.pdf
[88] "The Elements Project Confidential Transactions," https://www.elementsproject.org/elements/confidential-transactions/
[89] "Grin, the Tech," https://grin-tech.org/
[90] *See* Wilcox *supra* note 79.
[91] Giulio Prisco, Zcash Creator on the Upcoming Zcash Launch, Privacy and the Unfinished Internet Revolution (Aug. 30, 2016) https://bitcoinmagazine.com/articles/zcash-creator-on-the-upcoming-zcash-launch-privacy-and-the-unfinished-internet-revolution-1472568389.
[92] "Ring Signature," *Moneropedia*, https://getmonero.org/resources/moneropedia/ringsignatures.html.

than a consortium of identified but potentially corrupt or infiltrated parties). Privacy is guaranteed by neglecting to share any information about transactions with these validators except for the minimized amount of information necessary to prove scarcity and non-repudiation. Additionally, selective disclosure ensures that counterparties and third parties can be given visibility into the details of any particular transaction whenever the initiator wishes to be transparent or is compelled to be transparent by regulation or investigation.

## IV. Use Cases in which Public Consensus is Critical

There are many use cases or applications that can be created and deployed equally well on public or private blockchain networks. There are, however, certain use cases that can only achieve their full potential if they use a public and permissionless blockchain network. These use cases for which public consensus is critical, not coincidentally, also happen to be at the fundamental level of information systems: identity, security, and payments.

The most obvious use case in which public consensus is critical is in building *general purpose* decentralized computing networks—the decentralized computing platforms discussed at the start of this testimony. Just as the Internet has become a public platform for the proliferation of innumerable useful applications dealing primarily with communication of information, so too could networks like Bitcoin, Ethereum, Zcash, Monero, or Grin become platforms for innumerable applications dealing primarily with recordkeeping, exchange, and governance. The principle advantage of using public consensus mechanisms to form the basis of these platforms is the dynamism and diversity inherent in an open ecosystem of application developers, where developers need not seek permission to tinker with, create, and test a new idea.

But speaking abstractly of a variety of applications that will presumably emerge in a non-permissioned environment is not particularly satisfying. So for the remainder of this testimony we will discuss three specific, promising use cases that would particularly benefit from being built on top of public platforms.

The three use cases we will highlight can all be thought of as *applications*, a word we have thus far thrown about haphazardly without definition. By applications we mean *human jobs or problems that benefit from computing*. At the start of each subsection we will specify the specific human job or problem under discussion, and then go on to explain why that application would benefit from being built on top of a public consensus mechanism rather than a private and permissioned system.

A public consensus mechanism decentralizes trust, spreading out power on the network across a larger array of participants. In general, decentralization helps ensure **user sovereignty, interoperability**, **longevity**, **fidelity**, **availability**, **privacy**, and **political neutrality**. These attributes will be explained in the context of each application, and a discussion of public and private consensus mechanisms for that application will follow.

Speaking generally, however, and abstracting away some technical nuance, public consensus mechanism are critical in use cases where any of these attributes are desirable because only by including the user's device or an unbounded set of disinterested proxies for that user's interests in the consensus mechanism (by designing that mechanism such that *anyone* can participate and not just an empowered few) can the user free themselves from reliance on a single centralized counterparty to guarantee their privacy, the longevity of the network, the fidelity of the data in the blockchain, etc.

Again, public consensus mechanisms and the scarce tokens (like bitcoin or ether) that incentivize participation in the consensus, are not merely an artifact of the political biases of the initial creators of these technologies, they are also essential to the well-functioning of any system that desires user empowerment. So in the cases discussed below—electronic cash, identity, and the Internet of Things—we will explain why individual user empowerment is essential to the use case, and therefore, why public consensus mechanisms like proof-of-work or proof-of-stake are essential to building the infrastructure that powers those consumer or business applications.

### A. Electronic Cash

Bitcoin was the original blockchain and public consensus mechanism, and the white paper that first described the invention clearly describes the application it promised: "A purely peer-to-peer version of electronic cash [that] would allow online payments to be sent directly from one party to another without going through a financial institution."[93] Note that the design is more specific than often reported. Bitcoin was not designed to be a settlement tool for financial institutions, a lending or borrowing tool, a register for financial instruments, or a repository for any other sort of data. Bitcoin was designed to do one thing: enable cash-like (as in similar to paying with paper notes) transactions on the Internet.

#### i. What is Cash? Why is it Difficult Online?

Cash is a settlement tool, a very simple one that we tend to take for granted. Say I owe you $20 because you are a restaurateur who's just provided me with an excellent lunch. I have a debt that I can now settle very easily if I have cash: I hand you a $20 bill; done.

The peculiar utility of cash is derived from it being a fungible bearer instrument. A *bearer instrument* simply means that whoever holds the instrument is entitled to the rights described in the instrument.[94] The rights described by a $20 note were, historically, redemption by a bank or government of an equal amount in "real money" like gold coinage. The transition to fiat money altered that right subtly to redemption of any equally sized debt, public or private. In either case the possessor of the right is whoever holds the $20 note. *Fungible* means that any particular $20 note carries the same rights as any other $20 note (indeed two $10 notes

[93] *See* Nakamoto *supra* note 26.
[94] *See* William E Britton, "Transfers and Negotiations Under the Negotiable Instruments Law and Article 3 of the Uniform Commercial Code" 32 Tex. L. Rev. 153 (1953-1954).

together carries the same rights as well).

Fungible bearer instruments reduce transaction costs within any economic exchange.[95] In the midst of any given transaction, say paying the tab at a restaurant, neither party needs to pause and inquire as to the provenance of the note, whether it rightfully belonged to the buyer according to some authoritative registry of notes, or whether this particular note is blacklisted by virtue of being used previously in a crime or pledged as collateral in some ill-fated loan. Instead, the buyer presents the note, it looks like any other note, and would—as any other note—buy as much lunch. The transaction happens fluidly and without delay because the parties do not need to engage in fact finding or deep contemplation about the medium of exchange presented. Transaction costs are minimized. This particular reduction in transaction costs has long been understood as essential to a well-functioning economy. Take, for example, a report of the policy arguments made in a formative Scottish case on the subject of bank notes and fungibility in 1749:

> Policy issues, as might be expected, were highly prominent in Lord Strichen's Report. Trade, it was argued for the Banks, rested on the free circulation of money, and free circulation rested in turn on the reliability of notes and coins. If Crawfurd [the plaintiff, a previous holder of a bank note, and from whom the note in question was stolen] was able to vindicate the banknote, no merchant could risk taking money in payment 'without being informed of the whole History of it from the Time that it first issued out of the Bank or the Mint till it came to his Hand, which is so apparently absurd, that it seems hardly to merit a Consideration'. And as banknotes would thus be rendered 'absolutely useless', this would 'in a great Measure deprive the Nation of the Benefit of the Banks, which could hardly subsist without the Circulation of their Notes'. It was in vain for [opposing counsel] to object that, just as people continue to buy goods despite the (slight) risk that they might be stolen and subject to vindication, so they would continue to accept money if the risks were the same. If money could be vindicated, counsel for the Bank of Scotland concluded, 'no Man could be sure, that one Shilling in his pocket was his own, and … Banks might shut their doors.'[96]

Crawfurd lost his case and the fungibility of cash was guaranteed by the courts in Scotland. Similar decisions followed in other jurisdictions, and the fungible paper currency we know and rely on to this day was assured.

Compared with cash, pre-Bitcoin online transactions had relatively high transaction costs. This is because all electronic instruments are, effectively, registered instruments rather than bearer instruments. A *registered instrument* means that the rights associated with the

---

[95] *See generally,* David Fox, *Property Rights in Money*, §§ 2.11–2.20 (2008).
[96] *See* Kenneth Reid "Banknotes and Their Vindication in Eighteenth-Century Scotland" *University of Edinburgh, School of Law, Working Papers* (Nov. 2013) http://www.research.ed.ac.uk/portal/files/13523302/Reid_Banknotes.pdf. *quoting* Lord Strichen, Reporter, *Minutes, the Governor and Directors of the Bank of Scotland against the Governors and Directors of the Royal Bank and others* (21 February 1749).

instrument adhere only to the person whose name appears in some authoritative register, the current bearer of a particular certificate or note related to that instrument is irrelevant.

The reason why electronic instruments must be registered is straightforward. Digital files, like Microsoft Word documents or MP3 music files, can be costlessly duplicated. While the reproduction of a music CD will necessarily entail the costs inherent in the production of another physical thing, digital music files can be replicated with almost no effort or expense. If the bearer of a particular file is entitled to rights described in that file, and any person can almost costlessly copy the file again and again, then it is trivial to effectively manufacture more rights. A $10 file on my computer, if copied over and over can become a billion dollars. To address this, banks or other intermediaries will keep a centralized record (*i.e.*, a registry or ledger) of who has which rights to which electronic funds. If I claim to pay an online retailer, the retailer's computer effectively calls up my bank to make sure I have the money I say I do.

These registered instruments require mutual trust in the ledger keeper. If I'd like to pay you electronically, we'd both need to have an account at the same bank or else use an additional intermediary, like a correspondent bank or a credit card company, who can be a trusted go-between for our particular banks.

All of these intermediaries generate transaction costs. The magnitude of these costs will depend on the efficiency of the intermediaries, and the number of intermediaries necessary to build a trustworthy bridge between myself and the person I'm paying. Each may take a fee; each will take their time to process the transaction.

There are also hidden costs in these systems: chargebacks, and transactions forgone. Credit cards, for example, may appear to offer near instant transactions, but in reality the credit card company only *authorizes* future payment between the banks of the parties. If when that future payment goes to be settled (and even after the settlement), it turns out that the card has been reported stolen, the merchant receiving the payment may suffer a chargeback (*i.e.*, they will not receive the sum they were promised and they will bear the loss of the real goods they gave in exchange).[97] Additionally, when transaction costs are high, small-value transactions become cost-inefficient and people will simply avoid making them. This is the case with microtransactions to pay for or meter low-value digital goods (*e.g.*, a minute of Wi-Fi at the airport, the ability to read just one article on a pay-walled website).[98] Another substantial hidden cost is the unavailability of electronic payment to those who cannot obtain a banking relationship. Several billion people across the world do not have banking relationships, often through no fault of their own.[99] Banks will frequently deem a prospective customer's personal

[97] When goods are purchased using stolen credit cards, the merchant is generally left taking the loss. The Bureau of Justice Statistics estimates that these losses cost Americans over $24.7 billion in 2012 alone. That's 10 Billion more in losses than all other property crimes combine." *See* Bureau of Justice Statistics, Data Collection: National Crime Victimization Survey (NCVS) (2012) *available a*t http://www.bjs.gov/index.cfm?ty=dcdetail&iid=245.

[98] *See* Chris Smith, "What are Micropayments and How does Bitcoin Enable them?" *Coin Center* (June 2015) http://coincenter.org/entry/what-are-micropayments-and-how-does-bitcoin-enable-them

[99] Asli Demirguc-Kunt, Leora Klapper. Dorothe Singer, Peter Van Oudheusden, " The Global Findex

characteristics or the country where they reside as too indicative of risk for them to be profitable customers.[100] Women and other vulnerable groups are disproportionately affected by bank de-risking.[101] For these people, online transactions are simply not an option and the full global costs of these transactions-forgone goes uncounted.

### ii. Why Public Consensus is Critical for Cash

In a metaphysical sense, even paper bearer instruments exist on a "register" of sorts, but that register is global, decentralized, and easily made transparent. The register is the world of physical possession. Reading from the register looks like this: *whose hands or pockets hold which instruments?* And writing to it looks like this: *accept the note from the person who is handing it to you.* It is similar with bitcoin, but instead of hands and pockets and the physical world we have software and a global network. Bitcoin's key innovation was to *simulate* a bearer instrument digitally by using networked software to fully automate and decentralize the registry of instruments, such that the "registry" component of the instrument effectively fades into the background. My bitcoins are still described on a register and that's why I can't duplicate them willy-nilly, but the register is merely an unowned, shared, and ubiquitous feature of networked computers (just like the Internet is an unowned, shared, and ubiquitous communications feature for most computers today—and just like the ability to exchange paper notes or stuff them into wallets or safes is a ubiquitous feature of the physical world). When I transact with bitcoins I don't need to consider the blockchain or peer-to-peer networking

---

Database 2014 Measuring Financial Inclusion around the World" *World Bank Policy Research Working Paper* 7255 (April 2015) *available at* http://documents.worldbank.org/curated/en/187761468179367706/pdf/WPS7255.pdf#page=3.

[100] *See* Tracey Durner and Liat Shetret, "Understanding Bank De-Risking and its Effects on Financial Inclusion" *Oxfam Research Report* (Nov. 2015) *available at* https://www.oxfam.org/sites/www.oxfam.org/files/file_attachments/rr-bank-de-risking-181115-en_0.pdf. ("As financial institutions re-calculate risk appetites and decide to exit relationships, they directly and negatively affect these sectors and the populations they serve. For example, in August 2014, Westpac Banking Corp. followed other major Australian and UK banks and announced the closure of numerous money transfer operators' accounts over concerns about AML/CFT and rising compliance costs. This followed the precedent set in the wake of Barclays' May 2013 decision to close money transmitter accounts and the subsequent temporary injunction filed by Dahabshiil, one of the largest Somali remittance companies in the UK. The closure of these bank accounts not only threatens these businesses but also jeopardizes the vital flow of remittances to Somalia from diaspora populations, which constitute an estimated 25 to 45 percent of the country's GDP and serve as a key source of income for more than 40 percent of its vulnerable population.

[101] *Id.* at 6 ("For example, in developing countries, 46 percent of men have a bank account, compared to 36 percent of women. Immigrants are another heavily affected population: factoring out socioeconomic and demographic considerations, immigrants are six percent less likely to have a checking account and eight percent less likely to have a savings account in the US than their American-born counterparts. Without formal bank accounts, these underserved populations commonly rely on the remittance sector to send money to their families back home, and women have increasingly emerged as a key sending demographic. Although they remit about the same amount as men, women are shown to remit higher percentages of their income, more frequently, and for longer durations than their male counterparts. Reductions in the remittance sectors due to MSB account closures stand to further isolate these communities from the global financial system, exacerbating existing financial inclusion challenges.").

technology, just as when I visit a website I don't need to contemplate TCP/IP or HTTP.

To truly fade into the background, that system must exhibit certain qualities that real-world cash possesses:

**Some qualities exhibited by physical cash:**

- **User sovereignty:** The choice to initiate a cash transaction is entirely up to the person holding the cash. No intermediaries need be relied upon to ensure that the transaction can proceed.
- **Availability:** Cash transactions are always available. If you have cash on you, you can hand it to someone else.
- **Interoperability:** Within a given nation, everyone accepts and recognizes the value of cash. In the international context, the availability of liquid foreign exchange markets and the availability of a global reserve currency generally guarantees some level of global interoperability.
- **Longevity:** Cash has no expiration date, notes that have been hanging around in a mattress for years work just as well as fresh bills. Purchasing power may fluctuate over time but should not go to zero.
- **Fidelity:** Cash is designed to be difficult to counterfeit and to make counterfeit notes more obvious to the would-be recipient.
- **Political neutrality:** While the value of cash ultimately relies in part on its supply (a factor at least roughly controlled by governments and large banks) the ability to transact with cash is not contingent on any government or corporation. A holder of cash can hand that cash to another person without first seeking the approval of the issuing bank or government.
- **Privacy:** Cash transactions do not create a record.

**Electronic cash powered by a public consensus mechanism simulates these qualities:**

- **User sovereignty:** The bearer of a private key that corresponds to a pseudonym in control of some bitcoins is the only party able to initiate transactions and no particular transaction validator need be relied upon to ensure that the transaction can proceed.
- **Availability:** No particular transaction validator can block a user perpetually from transacting, nor would the technical failure of any particular validator stop the user from transacting because the process of writing and reading from the digital ledger is decentralized across a public network of peers, any of whom could serve as a validator.
- **Interoperability:** I don't have to have a common relationship with a particular validator and the person I'm paying in order to pay; all software necessary to utilize and interact with the network is freely available without seeking licenses or paying fees. While many may not immediately recognize the value of a bitcoin or other unit of electronic cash, the availability of liquid exchange markets generally guarantees some level of interoperability.

- **Longevity:** By decentralizing the storage of the ledger redundantly across all participants, and employing digital signatures to link all transactions into a unified data structure, the network ensures that even very old transactions never go missing from the ledger. Balances a user has left untouched for years or even decades are still available for spending.
- **Fidelity:** Transactions are recorded on the ledger in bundles called blocks. Transactions must obey logical rules to be incorporated into blocks (*e.g.*, spending the same bitcoins twice is not allowed). Transactions cannot be altered after the fact; any such attempted alteration would invalidate digital signatures within the block containing the transaction and in all subsequent blocks. These mismatched signatures highlight the fraud and (unless the full network of participants decide to change the network's rules against fraud) the attempt at alteration would be ignored. New transactions might be "erased" in favor of other transactions when one "block" replaces another within the most recent history of the ledger, but blocks further back in the ledger cannot be replaced without simultaneously replacing all blocks since that block, a process that would demand prohibitively costly computing resources.
- **Political neutrality:** By creating a public and global market for transaction validation and infrastructure upkeep, the network ensures that it would never be vulnerable to attempts by one government or institution to censor or stop particular transactions, or freeze particular balances. Additionally, the supply of the tokens is set by the software, and so would not be subject to the monetary policies of a state or the choices of a single corporation or institution.[102]
- **Privacy:** Bitcoin transactions *do* leave a record, but it is a pseudonymous record that generally does not make a user's full transaction history public information. The development of privacy-protecting technologies like zero-knowledge proofs or shuffling protocols may make identification of pseudonyms more difficult while also granting individuals the ability to selectively disclose information related to their transactions.

**<u>Private consensus</u> mechanisms would make it difficult to guarantee these features:**

- **User sovereignty:** The user must rely on the consortium members as intermediaries to ensure that the transaction will proceed.
- **Availability:** The identified members of the consortium could be compromised and the system could cease validating transactions or could be made to block the transactions of certain users. If the members collude they could block the transactions of certain users.

[102] Centralization of validators on an open network because of economic advantages from cooperation or geographic co-location is a real concern in these systems, however, thus far we've see little evidence of harms from this vulnerability. *See* Kyle Torpey, "Problems Associated With Bitcoin Mining Centralization May Be Overstated" *Bitcoin Magazine* (Sep. 2016) https://bitcoinmagazine.com/articles/problems-associated-with-bitcoin-mining-centralization-may-be-overstated-1474917259.

- **Interoperability:** Identified members could choose to only validate transactions from their collective customers, transactions between the users of one consortium's network and those of another may be more difficult or impossible.
- **Longevity:** The permanence of the balances on the ledger is guaranteed by the goodwill and the security practices of consortium members. If the ledger is not public, alterations or omissions could occur without scrutiny.
- **Fidelity:** Without a public ledger, users must trust the consortium members to vouch for the validity of any particular transaction history. Even if the ledger is regularly published by the consortium members and incorporates digital signatures, there is no process in place to reconcile discrepancies between the currently authoritative record endorsed by the consortium and some other version that, according to some users, proves that alterations have been made.
- **Political neutrality:** Consortium members retain the ability to censor transactions or blacklist specific funds, and censorship may be carried out for political purposes.
- **Privacy:** Transactions create a record that may or may not be pseudonymous. The privacy of this record is only guaranteed by the good faith and good technical practices of the consortium.

Only public consensus-driven networks can deliver the streamlining provided by true cash transactions. Instruments registered to a public blockchain can be treated as if they were bearer instruments because the process of updating the register is automated and decentralized: user sovereignty, availability, interoperability, longevity, fidelity, political neutrality, and privacy are effectively guaranteed by cryptography and economic incentives for honest participants.

If there is doubt about that automation, or if a set group of entities must be trusted to accomplish that purported "automation," the signed transactions cannot be treated as fungible bearer instruments. As in the case of credit card authorizations, we might fear repudiation if the automation is not guaranteed. As in the case of the unbanked, we might fear that some parties would be denied access to the system or have their transactions momentarily frozen because the trusted parties deem them too much of a risk. As in the correspondent banking context if the trusted parties refuse to make the register fully transparent or interoperable with other registers, we might fear that easy transactions can only be had between parties who have become customers of the same consortium.

Fundamentally, from a user perspective, a private blockchain technology doesn't "just work" from the get-go. I cannot send or receive money until I open an account and establish a legal relationship with a company. This may be a tolerable inconvenience, but it is not a system that works like cash, which can be accepted in the hand without any prior arrangements in place.

Only by fully automating the creation and maintenance of a ledger according to pre-established rules and economic incentives that play out in a public market for transaction validation can we be sure that electronic transactions are as good as cash.

### B. Identity

The Internet lacks a native identity layer. This shortcoming is the reason why Internet users must rely on a tapestry of weak passwords, secret questions, and knowledge of mother's maiden names to verify their identity to various web service providers. The need for a better solution is widely recognized,[103] and public blockchains may provide the answer.

#### i. What is Identity? Why is it Difficult Online?

In the physical world, identity is *federated*.[104] In other words, we don't have just one monolithic identity; we have a host of attributes. Nor do we have just one institution that vouches or attests that we have these attributes, we have several. A person's identity includes an endless variety of attributes: physical appearance, parentage and family history, citizenship, educational and employment history, skills, personality, etc. We seek and often carry evidence that others have attested to our attributes: driver's licenses, passports, birth certificates, membership cards, diplomas, letters of recommendation, professional certifications, awards, resumes, etc. In the physical world our identity is *user sovereign*: the bulk of these credentials are things over which we have immediate physical control; we keep them in our homes or our wallets; we might even wear them on our faces. We are in control of these attestations and can choose to show or decline to show them to others at will.

Online we should expect no different. As early as 1996, the need for robust digital identity systems was glaringly apparent. As the Clinton Administration noted in its Framework for Global Electronic Commerce:

> Of particular importance is the development of trusted certification services that support the digital signatures that will permit users to know whom they are communicating with on the Internet. Both signatures and confidentiality rely on the use of cryptographic keys. To promote the growth of a trusted electronic commerce environment, the Administration is encouraging the development of a voluntary, market-driven key management infrastructure that will support authentication, integrity, and confidentiality.[105]

But creating a robust, federated, and user-sovereign identity system that works online has proven difficult. As President Obama noted in a letter introducing the National Strategy for Trusted Identities in Cyberspace ("NSTIC") program:

> The rapid and vastly positive changes that have followed the rise of online transactions — like making purchases or downloading bank statements — have also led to new

[103] *See*, *e.g.*, Barak Obama, *Cover letter to the National Strategy for Trusted Identities in Cybersapce* (April 2011) *available at* https://www.whitehouse.gov/sites/default/files/rss_viewer/NSTICstrategy_041511.pdf.
[104] *See* Eve Maler and Drummond Reed, "The Venn of Identity: Options and Issues in Federated Identity Management" *IEEE Security & Privacy* (2008) *available at* https://css.csail.mit.edu/6.858/2012/readings/identity.pdf.
[105] *See* Clinton *supra* note 8.

> challenges. Few have been as costly or nerve wracking for businesses and families as online fraud and identity theft. These crimes cost companies and individuals billions of dollars each year; and they often leave in their wake a mess of ruined credit and damaged finances that can take years to repair. But there are other costs for our economy that are more difficult to measure. The potential for fraud and the weakness of privacy protections often leave individuals, businesses, and government reluctant to conduct major transactions online. For example, providing patients with access to their medical records from their home computers requires that hospitals be able to confidently identify that patient online.
>
> The simple fact is, we cannot know what companies have not been launched, what products or services have not been developed, or what innovations are held back by the inadequacy of tools, like insecure passwords, long overwhelmed by the fantastic and unpredictable growth of the Internet.[106]

One of the key challenges has been developing an interoperable system for online identity. As the NSTIC framework specifies:

> The third guiding principle of the Identity Ecosystem is to ensure policy and technology interoperability among identity solutions, which will enable individuals to choose between and manage multiple different interoperable credentials. Interoperability will also support identity portability and will enable service providers within the Identity Ecosystem to accept a variety of credential and identification media types.[107]

Interoperability is a technical challenge that demands a public, purpose-neutral platform through which users and institutions can present credentials and offer attestations depending on their particular needs. Researchers at Microsoft have stressed that:

> [D]ifferent identity systems must exist in a metasystem. It implies we need a simple encapsulating protocol (a way of agreeing on and transporting things) … The universal identity metasystem must not be another monolith. It must be polycentric (federation implies this) and also polymorphic (existing in different forms). This will allow the identity ecology to emerge, evolve, and self-organize. Systems like RSS and HTML are powerful because they carry any content. We need to see that identity itself will have several—perhaps many—contexts, and yet can be expressed in a metasystem.[108]

Another key challenge lies in creating a system that is privacy-protecting. As the NSTIC framework specifies:

> Just as there is a need for methods to reliably authenticate individuals, there are many

---

[106] *See* Obama *supra* note 103.
[107] *Id.*
[108] Kim Cameron, *The Laws of Identity* (May 2005) https://msdn.microsoft.com/en-us/library/ms996456.aspx.

> Internet transactions for which identification and authentication is not needed, or the information needed is limited. *It is vital to maintain the capacity for anonymity and pseudonymity in Internet transactions in order to enhance individuals' privacy and otherwise support civil liberties*. Nonetheless, individuals and businesses need to be able to check each other's identity for certain types of sensitive transactions, such as online banking or accessing electronic health records.[109]

This mirrors our discussion of privacy as contextual integrity. Depending on the circumstance, the user of the system should be empowered to control what identity information they reveal and what they keep secret. The goal of the system is, as was discussed in the context of zero-knowledge proofs, selective disclosure. Such a system cannot rely on perimeter security, obscuring private information by hiding it behind a firewall or using proprietary security software, in order to protect privacy. As researchers at Microsoft have stressed:

> Since the identity system has to work on all platforms, it must be safe on all platforms. *The properties that lead to its safety can't be based on obscurity* or the fact that the underlying platform or software is unknown or has a small adoption.[110]

Another key challenge has been creating a truly user-sovereign system. As the NSTIC framework stresses:

> Individuals shall be free to use an Identity Ecosystem credential of their choice, provided the credential meets the minimum risk requirements of the relying party[.] Individuals' participation in the Identity Ecosystem will be a day-to-day—or even a transaction-to-transaction—choice.[111]

Given these particular demands from online identity—interoperability, user sovereignty, and privacy—it should be increasingly apparent why public consensus mechanisms would be preferable in the development of online identity systems.

#### ii. Why Public Consensus is Critical for Identity

One way to look at Bitcoin is as a system that allows an otherwise anonymous individual to prove that they have a certain amount of funds without revealing any other personal details about themselves.[112] The same technology could be leveraged to prove all sorts of attributes

---

[109] *See* Obama *supra* note 103.
[110] *See* Cameron *supra* note 108.
[111] *See* Obama *supra* note 103.
[112] I can sign a statement that indicates I have control over some subset of my bitcoins, let's say 5. You can see that statement (or use software to read a verify it) and note that it is signed with the key that matches a public address on the blockchain, which has had 5 bitcoins sent to it in past transactions. I have proven that I control these 5. However, I may have other address that have more bitcoins. In this manner, a blockchain can be used to prove some limited facts about me without revealing more information about myself than I'd prefer. It is true that Bitcoin's blockchain currently leaks additional information about me, because clustering analysis may allow a stranger to determine the balances of all of my addresses (rather than only the address I've signed a message using) if my addresses have been

about an individual, effectively creating a user-sovereign, federated identity system.

Already some companies are experimenting with such a system. Today, for example, I can use a service called Onename, created by a company called Blockstack, to leverage the Bitcoin blockchain in helping me establish an online identity.[113] It works like this: I log into my Facebook account, my Twitter account, and my LinkedIn account and post a special message proving I control those accounts. A copy of that message is then signed with a digital signature that matches my established Bitcoin address.[114] Proof of those signatures can be encapsulated in the Bitcoin blockchain and the Onename website will make it easy for me to sign, write, and read those messages to and from the blockchain. Now, if I want to prove to someone who I am online, I can show them my signed messages on the blockchain and sign a personal message to them using the same key.

Effectively, the system allows the user to self-attest to an identity. The user shows that they have control over three different social networking profiles by creating signed attestations on each profile. A single Facebook account may be easy to fraudulently generate, but three different social media accounts, particularly if they have active use indicative of the person they purport to represent, would be harder to forge. With attestations from each account now available on the blockchain, we can be reasonably assured that any message signed with the private key matching that blockchain address is truly a message from the person who has those social media accounts.

We could imagine similar attestations from any number of federated attestors also residing as signed messages encapsulated and stored on the Bitcoin blockchain or any other public consensus blockchain. Now if want to prove I have a certain credit score, or a certain diploma, I can ask the credit rating agency or the university to sign an attestation and "transfer" it (as one would transfer bitcoins) to a public blockchain address I control. Now I can present that attestation, signing it again with my private key, to anyone curious about my creditworthiness or educational history. Because blockchains provide a sort of decentralized time-stamping, the attestation could be made to expire automatically, and subsequent on-chain messages signed by the attestor could revoke previous attestations if, say, my credit score changes or if my diploma is revoked.

These attestations could also be required of users who want to log into a given website, say an online banking account. Rather than mandating that a user create a password and use that password to log in, a bank could sign a login credential and assign it to that user's blockchain address. Now, to log in, she signs a login message with the private key that matches her blockchain address. The bank's website looks for that signed message, validates the signature,

---

used together in past transactions. This privacy weakness is, however, surmountable and, as discussed in the section on privacy (*see infra* at 35), several efforts are underway to make public blockchain networks more private, and capable of true granular information sharing and verification.

[113] *See* Ali *supra* note 5. *See also* https://onename.com/.

[114] *See*, *e.g.*, my personal Onename profile: https://onename.com/valkenburgh and an associated message I placed on my twitter profile: https://twitter.com/valkenburgh/status/595664205270880258.

and allows her to login. Reverse engineering a Bitcoin private key is effectively impossible, and that's a major step up from most user-set passwords that can be cracked in hours or even minutes by an enterprising hacker.

If the user loses her phone or laptop, her private keys could, of course, be compromised, and if she failed to keep backups she will be unable to sign messages proving her identity attestations. To solve this problem, public blockchain networks can leverage what are called multi-signature transactions. In essence, before accepting any attestation credentials at a given blockchain address, I empower three friends, co-workers, or institutions, with the ability to re-assign my credentials to another address should I ever lose my keys. Now if I lose my cell phone, I can call up my friends, ask them to revoke my credentials, and then meet with them to provision those credentials to a new address I've generated with the keys stored on my new device.

As with our discussion of electronic cash, it's now helpful to describe the key attributes offered by **public consensus mechanisms** and explain how they relate to an online identity system:

- **User sovereignty:** The bearer of a private key that corresponds to a pseudonym in control of certain identity attestations is the only party able to offer an attestation as proof of her identity, and no third party aside from the attestor who issued that attestation need be relied upon to ensure that the identification can proceed.
- **Availability:** No particular node on the network can block a user perpetually from offering attestations for identification purposes, nor would the technical failure of any particular node stop the user from offering attestations because the process of writing and reading from the digital ledger is decentralized across a network of peers.
- **Interoperability:** The user does not have to have a common relationship with any particular member of the network and the person to whom they are identifying themselves for an attestation to be shared; all software necessary to utilize and interact with the network is freely available without seeking licenses or paying fees. The user can seek attestation credentials from any individuals or institutions that choose to use the system and there is no fee or permission or establishment of any provider-customer relationship required for an attestor to join the system and start making attestations about users.
- **Longevity:** By decentralizing the storage of the attestations redundantly across all participants, and employing digital signatures to link all attestation transactions into a unified data structure, the network ensures that even very old attestations never go missing from the ledger. Attestations a user has left untouched for years or even decades are still available for proving her identity (provided they have not been set by the attestor to expire).
- **Fidelity:** Attestations are recorded on the ledger within transactions that are bundled into blocks. Transactions and their associated attestation data cannot be altered after the fact; any such attempted alteration would invalidate digital signatures within the block and in all subsequent blocks. These mismatched signatures highlight the fraud and the attempt at alteration will be ignored. New attestations might be "erased" when

one "block" replaces another within the most recent history of the ledger, but blocks further back in the ledger cannot be replaced without simultaneously replacing all blocks since that block, a process that would demand prohibitively costly resources in a proof-of-work or proof-of-stake consensus mechanism.

- **Political neutrality:** Attestation credentials are added to the system using the same transaction writing and transaction validation techniques employed by current bitcoin transactions. By creating a public and global market for transaction validation and infrastructure upkeep, the network ensures that it would never be vulnerable to attempts by one nation to invalidate attestations or revoke identities without the consent of the attestor.[115]
- **Privacy:** Writing attestations *does* leave a public record of a person's identity, but it is a pseudonymous record that generally does not make a user's full identity (all of her attestations) public information. The development of privacy-protecting technologies like zero-knowledge proofs or shuffling protocols may make identification of pseudonyms more difficult while also granting individuals the ability to selectively disclose information related to their identity (*e.g*, prove to a bartender that they are over 21, but avoid showing them irrelevant additional information such as name or address).

**Private consensus mechanisms would make it difficult to guarantee these features:**

- **User sovereignty:** The user must rely on the consortium members as intermediaries to ensure that attestations about them are made and incorporated into the system or shared with other users.
- **Availability:** The members of the consortium could be compromised and the system could cease offering access to attestations, or could be made to embargo the attestations possessed by certain users. If the members collude they could block the user from identifying herself to other users.
- **Interoperability:** Consortium members could choose to only permit attestations by certain institutions, and could forbid attestations to be made about their own customers. Identification verification between the users of one consortium's network and those of another may be more difficult or impossible.
- **Longevity:** The permanence of the attestations on the network is guaranteed by the goodwill and the security practices of consortium members. If the attestation data and associated digital signatures are not public, alterations or omissions could occur without scrutiny.
- **Fidelity:** Without a public record of attestations, users must trust the consortium

[115] Centralization of validators on a public network because of economic advantages from cooperation or geographic co-location is a real concern in these systems, however, thus far we've see little evidence of harms from this vulnerability. *See* Kyle Torpey, "Problems Associated With Bitcoin Mining Centralization May Be Overstated" *Bitcoin Magazine* (Sep. 2016) https://bitcoinmagazine.com/articles/problems-associated-with-bitcoin-mining-centralization-may-be-overstated-1474917259.

members as to the validity of any particular attestation. Even if the record of attestations is regularly published by the consortium members and incorporates digital signatures, there is no process in place to reconcile discrepancies between the currently authoritative record endorsed by the consortium and some other version that, according to some users, proves that alterations have been made.
- **Political neutrality:** Consortium members retain the ability to censor identity attestations, block user from asserting their identities, or blacklist specific users/identities, and censorship may be carried out for political purposes.
- **Privacy:** Writing attestations creates a record of users' identities. The privacy of this record is only guaranteed by the good faith and good technical practices of the consortium members.

In general, identity is a many-faceted concept. A person's identity is a bundle of qualities that she exhibits, and attestations that others make about her. If a centralized authority can see as well as revoke any and all of your credentials, it could present privacy and human rights issues. No such singular authority exists in the physical world where even a person denied a driver's license can still obtain a diploma, where a person denied a bank account can still get a passport, where the common infrastructure of identity is paper, plastic cards, or independent electronic records. We should expect nothing less from the digital world, and public consensus mechanisms are essential to that development.

### C. The Internet of Things

The promise of the Internet of Things is that every device you own or use—every "thing" in your home and beyond—will be "smart" and "networked."[116] From light switches to door locks, thermostats to toothbrushes, street lights to cars, everything will be collecting data about its use, will have a networked interface for remote usage, and will be able to communicate as needed with users or any other devices with which it may need to coordinate. Self-driving cars will whiz through intersections because their trajectories will be intelligently coordinated with other vehicles, refrigerators will know when you are running out of eggs or when the milk's gone bad and will order more, and every appliance in your home will be able to be switched off from hundreds of miles away if you're on vacation and worried you left something on.

Whether this utopian vision is likely or even desirable goes beyond the scope of this paper. Many homes already have smart thermostats, lights, door locks, televisions, and voice assistants like Amazon's Alexa, and even with these non-speculative, early-generation IoT devices, the need for public networks to underpin their operation is becoming apparent. Additionally, non-consumer, industrial IoT usage is on the rise. For example, smart devices can enable the automated monitoring of well-head flows across an oil field, equipment safety across a construction site, or soil moisture across a farm.[117] These uses also face the same security, availability, and longevity concerns as consumer devices but the consequences of

[116] *See* IBM *supra* note 58.
[117] *See* Saint-Andre *supra* note 56.

failure can be even more dire.[118]

### i. Why Public Consensus is Critical for the Internet of Things

IoT devices in general will need to identify themselves online for control and communications purposes. This means that all of the concerns we had about human identification in the previous section are again present with respect to device identification. IoT underscores the importance of decentralized identity because rather than merely being concerned with some 10 billion people who may each have multiple digital credentials (*e.g.*, *can drive*, *is over 18*, or *has credit score 729*) we must now also consider that each person may have 10 or even 100 smart devices in their home, business, or under their control, and each device may have multiple identities and credentials (*e.g. this lock can be opened by these five family members and this friend and these emergency personnel in case of an emergency*, or *this car must be capable of communicating with and then programmatically sharing the road with every other car that may be traveling today*). The sheer number of device identities and credentials inherent in projections of widespread IoT deployment necessitates that no one or handful of centralized authorities be in full control of that identification system. Reliance on one or a handful of identity validators would invite fragility into a massive and critical technological system; it would entrust reams of private data to a small group of actors who could engage in abusive or anti-competitive business practices or else become the target of devastating hacks.

Similarly, devices may need to shop and make payments. This is already the case for voice assistants like Amazon's Alexa, which can be used to shop for and buy consumer goods by voice interaction alone. This brings us back to several of the issues we encountered in the section on electronic cash. Payments, including device payments, should be under the control of the person whose value is at stake, the user. A device manufacturer need not retain the ability to block payments or accumulate private payments-related data merely because they sold you a piece of IoT hardware. A ride-sharing application developer should not necessarily retain the ability to limit your selection of possible drivers or prices merely by limiting the markets for drivers that your smartphone is capable of accessing. Consumer choice, privacy, and payment security can be bolstered if our connected devices can shop for us via decentralized markets powered by decentralized payment systems.

In previous sections we've looked at seven attributes of public consensus mechanisms and investigated how a particular use case may require these attributes. Rather than rehash all seven attributes here again, this section will focus on four that have particular importance in the IoT context: longevity, user sovereignty, privacy, and interoperability.

**Longevity**. A recurring annoyance for IoT pioneers (brave souls who have, say, already replaced all of their lightbulbs with smart bulbs) is unexpected or rapid "sunsetting" of a product by its manufacturer. This refers to a decision by the manufacturer to end technical or infrastructural support for the product. Within the realm of non-smart products, an end to

[118] *See*, *e.g.*, Kim Zetter, "An Unprecedented Look at Stuxnet, the World's First Digital Weapon" *Wired* (Nov. 2014) https://www.wired.com/2014/11/countdown-to-zero-day-stuxnet/.

manufacturer support can already be troublesome because customer service and repair may now become more difficult, but in the realm of smart products an end to support can be significantly worse.

A smart product will often only function properly when it is capable of connecting to and communicating with a server on the Internet that may, among other things, (A) help it identify itself and connect to other consumer products or Internet services,[119] (B) provide a web- or app-based user interface for the user to control the product's features,[120] and/or (C) store and process data essential to the device's operation.[121] That server will generally be operated and maintained by the device manufacturer and, should the manufacturer decide to take that server offline, the device may cease proper operation. This has been the case even with seemingly simple smart home products like light bulbs.

Take for example issues surrounding bulbs manufactured by Connected by TCP.[122] These bulbs were marketed as being compatible with other smart-home systems, in particular the Amazon Echo voice assistant (so that you could say, *e.g.*, "Alexa, turn on my kitchen lights")[123] and a mobile app called Wink that offers a dashboard for user control over a variety of smart devices (so that you would not need to navigate to various different apps on your phone to control devices made by different manufacturers).[124] The bulbs were also marketed as being capable of remote control over the Internet (so that you could turn them on and off even when out of the range of your home Wi-Fi network). Compatibility and remote control for the Connected by TCP bulbs was provided via a web server that was owned, maintained, and under the full control of Connected by TCP. The server would relay signals for switching the bulbs on and off from a user's Amazon Echo or Wink app to the user's Connected by TCP light bulb hub, and then, in turn, to the bulbs themselves.

In June of 2016, after years of selling these bulbs, Connected by TCP abruptly decided to take their server offline.[125] With the critical relay path to the bulbs now missing, all remote functionality and device interoperability disappeared. As a writer for Consumerist wrote:

> The bulbs still work as actual lightbulbs, if you want to use your lamp's on-off switch the old-fashioned way, and you can control them while inside the house on your home

---

[119] *See* Tobias Heer, *et al.*, "Security Challenges in the IP-based Internet of Things" *Wireless Pers Commun* (2011) *available at* http://link.springer.com/article/10.1007/s11277-011-0385-5.
[120] *Id.*
[121] *Id.*
[122] *See* Kate Cox, "TCP Disconnects "Smart" Lightbulb Servers, Leaves Buyers In The Dark" *Consumerist* (Aug. 2016) https://consumerist.com/2016/08/19/tcp-disconnects-smart-lightbulb-servers-leaves-buyers-in-the-dark/.
[123] *See* Michael Garcia, "Using Alexa Skills Kit and AWS IoT to Voice Control Connected Devices" *Amazon Developer* (May 2016) https://developer.amazon.com/blogs/post/Tx3828JHC7O9GZ9/Using-Alexa-Skills-Kit-and-AWS-IoT-to-Voice-Control-Connected-Devices.
[124] "Wink Hub" *Wink.com* http://www.wink.com/products/wink-hub/ *last accessed* Dec. 2016.
[125] *See* Cox *supra* note 122.

> WiFi network. But any remote functionality—a big part of the steep price tag that makes TCP bulbs more expensive than a plain old LED bulb—is long gone.
>
> The fact that the bulbs are still on store shelves, with packaging promising features that no longer exist, is irksome. But it's also not an uncommon tale in these early years of the Internet of Things. Businesses try, and then discontinue, new products all the time.[126]

The Federal Trade Commission has taken a careful look at this burgeoning problem, launching an investigation into Google's choice to end support for products manufactured by Nest, a smart-home firm it acquired.[127] The FTC ultimately closed that investigation but warned manufacturers of their concern over two key policy issues:

> First, there are serious issues at play when consumers purchase products that unexpectedly stop functioning due to a unilateral decision by the company that sold it. Consumers generally expect that the things they buy will work and keep working, and that includes any technical or other support necessary for essential functioning.
>
> Second, when a company stops providing technical support, including security updates, for an IoT device, consumers may be left with an out-of-date product that is vulnerable to critical security or privacy bugs. This could create vulnerabilities for other systems connected to these IoT devices, and put consumers' sensitive data at risk. And if hackers can hack a smart car, pacemaker, or insulin pump, the risks are even more serious.[128]

Public consensus mechanisms can provide significantly enhanced longevity by replacing a privately owned and maintained server with a decentralized computing network. Device identity and data storage can be be offloaded to a decentralized ledger and decentralized file system and the device can even be pre-loaded by the manufacturer with a modest amount of funds to pay the global network of parties contributing resources to that decentralized network for the device identity registration, data storage, and connectivity that it needs for a reasonable lifetime. Now, even if the manufacturer goes out of business, if it decides to change its product offerings, or is acquired by a company unwilling to continue device support, the device itself will continue to have the same network infrastructure necessary to maintain proper functioning.

A private consensus mechanism may not provide this guarantee of longevity. The consortium members, just like the company with a centralized server, may choose to deprecate support for older products, or they may shut down the network entirely. Only a public network where participants are free to come and go and are incentivized to participate by device payments

---

[126] *Id.*
[127] Jessica Rich, "What happens when the sun sets on a smart product?" *FTC Business Blog* (Jul 2016) https://www.ftc.gov/news-events/blogs/business-blog/2016/07/what-happens-when-sun-sets-smart-product
[128] *Id.*

will assuredly continue to function for as long as devices continue to pay. Additionally, if the device's on board wallet is pre-loaded with electronic cash powered by a public blockchain network, then reloading the device with new funds is a simple process that anyone in possession of the device (perhaps even after multiple resales) could accomplish.[129]

**User sovereignty and privacy.** Nobody wants a baby monitor, security camera, or even a remote-activated light bulb that several dozen complete strangers may be able to access and control. In the world of "dumb" devices this was easy for a device designer to avoid: unless you have physical access to the switches on the device, you have no control over its operation. So a baby monitor that is closed-circuit or that only broadcasts analog signals will generally be in the sole and sovereign control of people in the house. Assume there are locks on the doors and we have good user-sovereignty and privacy.

Smart, internet-connected devices, however, when they rely on web servers for their functionality, will often fail to have these qualities. Recall Nick Szabo's characterization of the web's client-server architecture:

> When we currently use a smartphone or a laptop on a cell network or the Internet, the other end of these interactions typically run on other solo computers, such as web servers. Practically all of these machines have architectures that were designed to be controlled by a single person or a hierarchy of people who know and trust each other. From the point of view of a remote web or app user, these architectures are based on full trust in an unknown "root" administrator, who can control everything that happens on the server: they can read, alter, delete, or block any data on that computer at will.[130]

This applies to any device in the home that connects to the Internet as well as it does to a smartphone or laptop. Let's imagine a baby monitor that can be switched on and off remotely, and that broadcasts audio and video to the user's smartphone. Generally, these devices are manufactured to use a client-server architecture.[131] The logic of the application (rules for how and when the device should turn on, rules for who has access to the device, rules for how data from the device should be routed) exists on a server controlled and maintained by the device manufacturer and physically remote from the device (probably in a large data center somewhere).[132]

The user connects the baby monitor to the Internet using the home's wired or Wi-Fi connections and the device, in turn, connects to the manufacturer's web server; the baby monitor is now one client of the server. The user then sets up her smartphone with an app provided by the manufacturer for controlling the baby monitor and viewing the feed. The user's device is *another* client of the server. When the user decides to switch on the monitor from her cell phone, a message is sent to the server, checked for authenticity, and then relayed

---

[129] *See infra* at 45.
[130] *See* Szabo *supra* note 2.
[131] *See* Heer *supra* note 119.
[132] *Id.*

to the device itself. The baby monitor turns on. Unlike a light switch that completes a circuit entirely within the home, this "circuit" exists across potentially hundreds of miles of Wi-Fi, cellular signal, satellite, fiber-optic cable, and server warehouse. Similarly, when the baby monitor relays a video feed of baby, that data travels back across the Internet, to the server, and then back to the user's device (this may be the case even when the user is in her own home and near the monitor).

This system architecture presents a major issue from a user-sovereignty standpoint. Unless the application server is very carefully designed, someone with physical access to that server may be able to control the baby monitor as easily as the user can from her cell phone. Indeed, if the application server is poorly designed (*e.g.* firewalls are not well employed, user passwords are not strong and properly stored, encryption is not used to mask data coming and going from the server, and/or streaming protocols are employed without password-protection) then anyone in the world with an Internet connection may be able to control the baby monitor.

This is not as rare of problem as it may sound. Indeed, there is a search engine, Shodan,[133] that can be used to comb the Internet for connected devices that promiscuously broadcast unprotected video feeds, as reported by Ars Technica:

> Shodan, a search engine for the Internet of Things (IoT), ... includes images of marijuana plantations, back rooms of banks, children, kitchens, living rooms, garages, front gardens, back gardens, ski slopes, swimming pools, colleges and schools, laboratories, and cash register cameras in retail stores, according to Dan Tentler, a security researcher who has spent several years investigating webcam security. "It's all over the place," he told Ars Technica UK. "Practically everything you can think of."[134]

Off-loading as much device registration and application logic as possible to decentralized systems should provide enhanced user-sovereignty. This may be relatively straightforward when it comes to authentication. As discussed in the section on identity, the user can provision herself (*e.g.* her smartphone) and the smart device with identity credentials and access rules that would reside on the blockchain. The device can always query the blockchain for a current list of authorized users (*e.g.*, pseudonyms that must sign with matching private keys to gain access) and users can rely on multi-sig setups to revoke credentials if their smartphone is lost or stolen.

Data from the device, say video feeds from a security camera, can be encrypted and stored locally or in a decentralized file system[135] where members of the network provide surplus storage in return for payments from devices. So long as the keys to the encrypted data remain

---

[133] Shodan, https://www.shodan.io/ *last accessed* Dec. 2016.
[134] J.M. Porup, *"'Internet of Things' security is hilariously broken and getting worse"* ArsTechnica (Jan. 2016) http://arstechnica.com/security/2016/01/how-to-search-the-internet-of-things-for-photos-of-sleeping-babies/.
[135] *See*, *e.g.*, IPFS, https://ipfs.io/ *last accessed* Dec. 2016.

with the user, none of these otherwise anonymous storage providers will be able to access or view the encrypted files.

Computing tasks that the device may need to perform in order to function, say analyzing video data to find human faces or identify intruders, can be designed to run locally on the device only, rather than on a server. Alternatively, those computing tasks could also be offloaded to a decentralized computing network[136] where participants offering computing services are rewarded by payments from the device for data processing. In this case, of course, no private data should be shared with the decentralized network unless it is encrypted. This may appear to limit the value of a decentralized computing network: how can the network process the data if it cannot view it unencrypted? The science of distributing computing work amongst several participants without fully revealing encrypted data data is a vibrant and growing subfield within cryptography, generally referred to as *secure multiparty computation*.[137]

One technique in this field is the development of robust *homomorphic encryption*,[138] which means that a computation performed on an encrypted file will yield the same result as a computation performed on a plain text (not encrypted) file. So in our video analysis example, the decentralized network can still process the video data and give a result: *in this 12 hours of video there was one human intruder who entered the house*, but the various maintainers of the several computers that may have been involved in that decentralized data processing cannot ever see the unencrypted video file and therefore cannot ever see any details about the device-user's home (aside from knowing that there was one human intruder within a given time, as per our example).

*Zero-knowledge proofs* provide another cryptographic tool used to achieve this level of privacy.[139] As described previously, a ledger of transactions can be effectively encrypted or hidden but a zero-knowledge proof can still process the data in that ledger and reveal whether any transactions attempted to double spend funds. In this sense a public ledger can still be privacy protecting while still guaranteeing that all transactions were valid and not counterfeit. This can work in the IoT context as well. Rather than "all transactions were valid," the limited proof is "all smart lock door openings were from authenticated users," and only this data becomes public not the specific times that the door was opened or the identities of the authorized lock openers.

Another tool to build these system architectures is the division of computational work into several small pieces and the assignment of that work across several unaffiliated participants none of whom can see the entire file being processed and, therefore, see the private data undergoing computation. The Enigma Project out of MIT is an effort to build just such a secure

---

[136] *See*, *e.g.*, Ethereum, Buterin *supra* note 6.
[137] *See* Yehuda Lindell and Benny Pinkas, "Secure Multiparty Computation for Privacy-Preserving Data Mining" *Journal of Privacy and Confidentiality* (2009) *available at* http://repository.cmu.edu/cgi/viewcontent.cgi?article=1004&context=jpc
[138] *See id.* at 79.
[139] *See id.* at 76.

multi-party computation system that relies on a blockchain to divide work into pieces, keep track of the pieces, find participants, and assign work among them.[140] This avoids reliance on a single trusted intermediary to achieve the division, a potential vulnerability if that intermediary can reassemble the pieces and see the private data being processed.

In general, the computation, data storage, and network access rules currently found within a server-client architecture for smart home devices could be decentralized by using public consensus mechanism driven networks. In theory, a private consortium driven network could achieve similar results. However, this reintroduces trust in the identified members of the consortium, weakening the goal of pure user-sovereignty.

**Interoperability.** Smart devices need to interact with other smart devices. The door sensor needs to communicate with the smart bulbs in order to make the hall lights come on if you come home after dark. Self-driving cars need to communicate with other self-driving cars if they are going to have smart collision avoidance and traffic pattern automation. An Amazon Alexa or similar voice controlled assistant needs to communicate with digital music retailers in order to let you shop for new music by voice.

Herein lies, perhaps, the most common sense argument for using public consensus mechanism networks to power devices in the Internet of Things. If the infrastructure powering a smart device is owned and controlled by one particular manufacturer, integrating that device with other devices may be difficult. Worse, that integration may be made deliberately difficult to gently cajole the customer into buying all of their devices from one manufacturer. This is the issue of so-called *walled gardens* in computing systems: everything is beautifully manicured but you can't leave.[141] If customers cannot choose competing products without suffering the substantial switching costs inherent in replacing *all* of their IoT devices, free and open competition suffers, and prices rise.[142]

This is particularly the case with devices that deal with online shopping. Take Amazon Echo for example. This voice assistant allows the user to order products merely by asking for them. Simply say, "*Alexa, buy me some cat litter!*" and the device will look at your past shopping habits, propose a brand, amount, and price, and allow you to agree or ask for another option. There is a fascinating and undeniably convenient feeling associated with truly hands free shopping.

But, of course, having an Alexa in your home will mean you are locked in with one retailer, Amazon, for any and all hands-free shopping that you do. When Alexa queries your shopping history and the varieties of cat litter on offer, she only shops Amazon's suppliers and partner merchants. Similarly, if you ask Alexa to play music, she will only be able to play songs you

---

[140] Guy Zyskind, Oz Nathan, Alex "Sandy" Pentland, *Enigma: Decentralized Computation Platform with Guaranteed Privacy*, (Dec. 2015) http://www.enigma.co/enigma_full.pdf

[141] *See* Richard Firth, "Beware the walled gardens" *itWeb Open Source* (Mar. 2013) http://www.itweb.co.za/index.php?option=com_content&view=article&id=62788.

[142] *See* Carl Shapiro & Hal r. Varian*, Information Rules: A Strategic Guide To the Network Economy* 109-10 (1998) (discussing strategies to deter customer mobility by imposing switching costs).

bought or uploaded to your Amazon account; she can't play from the collection you've amassed on, for example, iTunes. Ideally, a device would be able to access any of the digital property the user has previously purchased, and it should comparison shop across all willing sellers for things the user has yet to buy, selecting the best price for the item she wants. This open competition can only be achieved if the markets for buying and selling are truly decentralized.

Several firms are building the tools to accomplish just such decentralized commerce; one that warrants highlighting in this testimony is OpenBazaar.[143] OpenBazaar is, in essence, a decentralized eBay where buyers and sellers can find each other and engage in a safe exchange. Buyers and sellers are protected from fraud on OpenBazaar by leveraging multi-sig bitcoin transactions to place funds in a sort of trust-minimized escrow while goods are in transit or being evaluated for quality. In the event of a dispute a neutral third party arbitrator is invoked who can redirect the funds to either the seller or the buyer based on their decision regarding who was in the wrong in the disputed transaction. Additionally, OpenBazaar uses BlockStack's decentralized identity tools to create and authenticate the identities of buyers and sellers, and may soon use a decentralized files system, IPFS,[144] to host images and descriptions of items listed for sale. The result is an online shopping experience just like eBay, but it can exist on decentralized network where there is no company like eBay that has any control over the sales that occur on their platform.

There is not a good case for using regulation to force device manufactures to participate in public decentralized markets; walled gardens can have their appeal and regulations can have unintended consequences. However, it's important for policymakers to understand the potential value decentralized networks provide in fostering open digital exchange and commerce that could be foundational to better, future IoT systems.

Altogether, the case for having public consensus mechanisms power IoT blockchain networks is clear and linked to our prior discussion of identity and electronic cash. First, public blockchain networks allow for a truly decentralized data-structure for device identity (I am a bulb in this home) and user access authorization (user with address 0xE1A… is the only person who can turn me on and off). The redundant and decentralized nature of data on these networks can ensure that these systems have true longevity, and a manufacturer's decision to end support for a product will not destroy the user's ability to securely access the product's features. Second, public blockchain networks can ensure that devices are interoperable and compatible because critical infrastructure for device communication, data storage, and computation can be commoditized and shared over a peer-to-peer network rather than be owned (as a server warehouse is owned) by a device manufacturer that may be reticent to opening its costly platform to competitors. Third, device payments for supporting and maintaining that networked infrastructure or allowing the device's user to easily engage in online commerce can be made efficient by utilizing the electronic cash systems that only

[143] OpenBazaar, https://openbazaar.org/ *last accessed* Dec. 2016.
[144] IPFS, https://ipfs.io/ *last accessed* Dec. 2016.

public consensus mechanisms can facilitate.

## V. Conclusion

*All* new approaches to decentralized computing—whether private or public—should be celebrated and allowed to develop relatively unfettered by regulatory or government policy choices. Much as the Clinton Administration took a light-touch approach to the development of the Internet in the 1990s, so should policymakers approach these new systems, however designed.[145]

In order to make good policy choices and ensure that the U.S. remains competitive in a global technological market we need a more detailed and productive discussion of these new tools. We need a basic understanding of how consensus works, what it might help us build, and why public and pseudonymous networks, despite their easily apprehended risks, offer significant and otherwise unattainable benefits. This testimony has offered a non-technical explanation of key variables within consensus mechanism design, catalogued why public mechanisms may, for certain use cases, be more worthy of user trust and more capable of ensuring user privacy and security.

The benefits of this technology are real. Electronic cash promises efficient microtransactions and enhanced financial inclusion; robust digital identity may solve many of our online security woes and streamline commerce and interaction online; and blockchain-driven Internet of Things systems may spur greater security, competition, and an end to walled gardens of non-interoperability for connected devices. However, our three highlighted use cases are likely only the tip of the iceberg. Just as few would have predicted the emergence of Facebook or Uber given only an understanding of the Internet circa 1995, it is impossible to know what creative and diverse minds will build when offered a free and public platform for experimentation.

[145] *See* Clinton *supra* note 8.