GeeqChain
A Better Approach to Blockchain

John P. Conley[1]

Version 1.25

January 2018

Abstract

We propose a scalable, inexpensive, and computationally light approach to building blockchains called the Catastrophic Dissent Mechanism (CDM). The CDM is based on game theory and economic mechanism design and takes a completely new approach to transaction validation: Proof of Honesty (PoH). The network of validators constructs blocks that contain enough data for users to independently verify that the chain is "honest" in the sense that it contains only valid transactions and that the chain as a whole follows protocol. This permits a system of audits that gives all actors strong incentives to behave honestly, both as individuals, and as members of coalitions who might benefit by compromising the integrity of the blockchain. As a result, the CDM offers users Strategically Provable Security (SPS), in effect, 100% Byzantine Fault Tolerance (BFT). The protocol also allows for the creation of an ecosystem of federated chains that can safely share GeeqCoins and other native tokens, but which can support many different types of internal business logic. This makes GeeqChain suitable for a wide variety of use cases, allows for upgrades and bug fixes without breaking protocol or instituting hard forks, and offers an alternative to Ethereum's ERC20 standard for startups wishing to create new platforms to take advantage of blockchain's potential.

[1] I would like to thank Stephanie So of Vanderbilt University for her contributions in developing the ideas in this paper, Ric Asselstine of Terepac Corporation for his vision and work to bring these ideas into reality, and Yorke Rhodes of Microsoft for turning my attention to blockchain and cryptocurrencies in the first place.

jpc@jpconley.com
www.jpconley.com

1. Introduction

Traditional databases are maintained on private servers by central authorities who control access, grant permission to alter, update and delete records, and who are ultimately responsible for the accuracy of the data. Trusting such data is equivalent to trusting in both the honesty and the security competence of the central authority.

Blockchains are append-only, distributed ledgers. No central authority owns or controls the data. Users send requests to write new records to a set of decentralized, often anonymous, nodes[2] who must come to a consensus on their validity. Once a record is written to a block and committed to the chain, it becomes both immutable and nonrefutable. The cryptographic signatures and recursive hashing of blocks make it impossible to delete, alter, or claim that one never agreed to the contents of a record. If a blockchain is public and transactions are written in cleartext (as they are in Bitcoin), records in the chain can be independently verified by any user who wishes to do so. If copies of the chain are stored in many places, it becomes almost impossible to censor or prevent access to the data it contains.

Blockchains allow agents to cooperate without the need to trust in the honesty or good behavior of one another or any third party. For example, Bitcoin's transaction protocol ensures that a sender has enough tokens in his account to cover a transfer and, once the transfer is made, the receiver can be secure in the knowledge that it cannot be reversed. Ethereum's smart contracts permit even more sophisticated interactions between users without the need for mutual trust.[3]

Unfortunately, the promise that blockchain holds for creating decentralized and trustless ways to share information and improve distributed business processes is limited by several factors. Some approaches (Bitcoin and Ethereum, for example) have high transactions fees that are a consequence of their security models. Others (such as Hyperledger fabric, and Iota's tangle protocol) have lower transactions costs, but offer lower levels of security, have central points of failure, or depend on some degree of trust in the validators. The ability to scale to large numbers of transactions per second, fix bugs, and upgrade to better and more flexible protocols is also limited on existing blockchain platforms.

GeeqChain offers a new approach to blockchain based on game theory and economic mechanism design. GeeqChain is secure, cheap, fast, and scalable. It can be implemented with fully anonymous verifying nodes, no centralized points of trust or failure, and any level of encryption and

[2] Nodes are computers on a network that participate in validating transactions and writing blocks of transactions to the chain. Depending on the blockchain and approach to validation, nodes are called miners, stakeholders, delegates, or voters, among other terms. Humans (who we call agents) own these computers and make them available to the network. One agent may be the owner of several nodes, or may simply be a user who makes transactions on the blockchain but does not provide validation services.

[3] It should be noted, however, that the Ethereum smart contracts have also led to a number of significant security issues. For example, on June 17, 2016, a coding error in the smart contract supporting the DAO resulted in a theft of 3.4M ETH worth $53M. More recently (November 6, 2017), a coding error in the Ethereum smart contract supporting Parity's multi-signature cryptocurrency wallets locked up accounts holding over 500k ETH worth over $150M.

privacy protection desired. The protocol allows the creation of an ecosystem of federated chains that can safely share GeeqCoins, but which can support very different types of internal business logic. This makes GeeqChain suitable for a wide variety of use cases, allows for upgrades and bug fixes without breaking protocol or instituting hard forks, and offers an alternative to Ethereum's ERC20 standard for startups that wish to create new platforms to take advantage of blockchain's potential.

2. Outstanding Problems with Existing Blockchains

Blockchain has the potential to profoundly transform the way we work, transact, and share information. As a technology, however, blockchain is immature and several significant problems need to be solved before this potential can be achieved.

2.1 Security

All the advantages that blockchain offer depend on honest transaction verification and block-writing. Bitcoin and Ethereum use a network of "miners" and a Proof of Work (PoW) protocol to establish the integrity of their ledgers. In the case of Bitcoin, block-writing rewards and transactions fees create incentives for honest behavior. Provided that more than 50% of the miners are moved by these incentives, the chain is difficult to corrupt.[4] As a result, the Bitcoin protocol is said to have a Byzantine Fault Tolerance (BFT) of 50%[5]. We discuss the value of BFT as a measure of security in more detail below.

In PoW protocols, nodes are generally run by anonymous agents. The principle of one CPU, one vote, applies. Any agent who is willing to bear the computational cost of trying to mine a block can join the validation network anonymously and as an equal. The hope is that the computational cost deters Sybil attacks in which many "fake" identities are created in order to gain majority control of the validation process. If votes must be paid for with work, then it should be too costly to mount such an attack.

In practice, many Bitcoin and Ethereum nodes are owned by the same real-world agent or are part of mining pools in which hardware may be owned by different agents but which coordinate their efforts and share rewards. Many pools have chosen to self-identify which makes them vulnerable to pressure from state-actors or others. Mining pools are so concentrated at this point that if no more than three were to collude, they could mount a successful 51% attack. In effect, Bitcoin is not validated by thousands of independent nodes but depends instead on the honesty of three or fewer agents. Put another way, if Bitcoin has 10,000 nodes, an attack by only three agents would be successful. In a real sense, this means that the BFT of Bitcoin is only 3/10,000 or .03%.

[4] More accurately, if miners controlling more than 50% of the network's hashing power are honest, then the chain is difficult to hijack. If more than half are dishonest, they will end up writing the majority of the blocks and this gives dishonest miners effective control over what the blocks contain (and the ability to create the longest, and therefore, the only valid chain under Bitcoin protocol). This is called a "51% attack".

[5] However, Eyal and Sirer (2014) describe a type of coordinated attack by miners holding only 25% of Bitcoin's hashing power that can compromise the blockchain implying that Bitcoin is only 25% BFT.

One must wonder why these pools do not, in fact, merge. In any industry, mergers create market power. For example, if there are four firms selling cellphone service, each making $X per year in profits, the monopoly created by merging all four firms would make more than $4X in profits. At worst, the merged firm could simply proceed as if it was still four separate firms and make exactly $4X. Taking advantage of monopoly pricing or economies of scale, however, would certainly bring profits above $4X. Merging is in the interest of the shareholders of all four firms. In the same way, if three mining pools are each making $X in net profits from mining rewards and transactions fees, the merged pool could make at least $3X by continuing to act as they did before. On the other hand, the merged pool would be able to mount a successful 51% attack and take over the Bitcoin blockchain, gaining whatever additional profits this might entail. The fact that they do not choose to do so must mean that something besides the PoW protocol is keeping them honest. The most likely candidate is the fear that stealing bitcoins would result in a hard fork[6] that would prevent the merged pool from profiting from their theft. In other words, to the extent that PoW blockchains are trustworthy, it is only because of the belief that "code is law" is a lie. It is in fact the threat to break protocol, not the protocol itself, that keep Bitcoin and Ethereum safe.

Proof of Stake (PoS) is the other main approach to verification. Several banks, for example, might set up a private blockchain in which the members vote on whether a new block is correct and should be added to the chain. Stake can also be established by posting a bond (money, tokens, work, provision of resources such as storage or bandwidth, participation in network activities, etc.) and voting power is then distributed in proportion to the stake. It is possible to allow free entry of anonymous validators in these cases.[7] Depending on the implementation, PoS approaches are both cheaper and more scalable than PoW. Unfortunately, PoS depends on the majority of stakeholders behaving honestly. The number of voting stakeholders (tens or hundreds) is typically much smaller than the number of validating nodes (tens of thousands) used by PoW blockchains. This makes collusion by validators much more likely and so it is not clear how much confidence a claim of 50% BFT should give us.[8]

[6] In this case, the "hard fork" would involve breaking protocol and ignoring the "validated" blocks containing transactions that were judged as stealing coins. New blocks would be added starting from the last "honest" block. More generally, a hard fork takes place when part of a group decides to take a project in a new direction starting from the existing code base, dataset, or other IP. This is usually because the group has a different vision of the best path forward. Hard forks are especially problematic in context of blockchain because it breaks the rule that "code is law" by creating a fork with new laws and protocols. Even if the motivations are pure, if you can break the law for good, you can break the law for bad. Hard forks are very corrosive to the trustless, anonymous, distributed nature of blockchain. See footnote 23 for some additional discussion.

[7] There are also many hybrid approaches that use combinations of PoW and PoS or even more complicated means of choosing agents to verify transactions.

[8] If a fixed set of stakeholders validate a blockchain, then honest behavior depends on the incentive structure faced by these specific agents. For example, one might be confident that the reputational damage of dishonest behavior would be enough to make Bank of America or Deutsche Bank behave correctly. When voting stakeholders can choose actions that affect their voting power, however, dishonest agents, who have the most to gain from subverting the blockchain, have the greatest incentive to expend the effort required. Thus, protocols that use escrowed tokens or Proof of Effort of some kind may end up systematically choosing dishonest validators. BFT loses its meaning as a measure of security in such cases.

For example, in a private blockchain created for a small group of banks, it is seen as unlikely that any of them would sacrifice their reputations in order to falsely verify transactions. It is seen as even more unlikely that 5 out of 8, or 26 out of 50, would simultaneously become dishonest.[9] There are two problems with this. First, as above, if the identity of the validators is known, they can be pressured by state-actors to break validation protocol in support of legal judgments or state policy. Second, it may very well happen that several banks merge or find they are in financial difficulty at the same time[10] It is not hard to imagine the possibility of collusion by a majority of validators in such a case.

Even if stake-holders are numerous and anonymous, we run into the same concentration problem that we see in PoW protocols. If the profit a validator gets from posting a bond is worth it, why not post the same stake under many identities? At worst, each identity makes enough profit to pay for the cost of posting the bond. Creating enough identities to gain the majority of the total voting stake makes it possible for a single real-world agent to take over the blockchain. Again, it is only the threat of out-of-protocol actions that creates a disincentive to make such an attempt.

2.2 High Cost

Visa and Mastercard charge merchants a fee of about 25¢ plus 2.5% of the value of the transaction to use their networks. These transactions costs are very high, certainly too high to make it practical for a customer to make a micropayment of a few cents to a merchant or content provider. One of the great promises of blockchain-based cryptocurrencies is that they will make financial transactions more efficient. If Bitcoin, Ethereum, or any of the other alt-coins now in existence found a way to allow people to make transactions quickly, cheaply, and securely, it would revolutionize the financial industry.

The Ethereum and Bitcoin platforms have transactions costs that range from tens of cents to tens of dollars. Blockchains that depend on PoW protocols must have very large networks of validating nodes. Users ultimately bear the costs of having thousands of nodes using electricity and wasting CPU cycles to solve the cryptographic puzzles required, to win block rewards and validate transactions.[11] In other words, high transactions costs are baked into PoW based cryptocurrencies.

[9] Of course, Proof of Stake begs the question of why a blockchain is needed at all. If a group of firms mutually trust one another, why is there a need to create an immutable record validated by a complicated PoS algorithm?

[10] Since all the participating banks are in the same sector, their economic fortunes are highly correlated. In a recession or financial crisis, all banks are likely to be under financial pressure, the threat of bankruptcy, and the possibility of being taken over by the federal reserve. It would not be at all surprising if a financial crisis, such as the one that began in 2007, were to result in five out of eight banks in a PoS blockchain being placed under federal supervision or forced to merge. Bank officers might be willing to take desperate measures to survive. The threat of a lost reputation is not much of a deterrent to a bank or any firm facing extinction.

[11] The bitcoin protocol creates a cryptographic puzzle for miners to solve at the beginning of each block. This puzzle can only solved by brute force, trial and error. Miners run computers (more often, large clusters of purpose built computers) to find the solution. Solving the puzzle produces what is called a “nonce”. Once the nonce is found, it can be used by anyone to quickly verify that the puzzle has been correctly solved. It is estimated each block requires that 10^{25} “hashing operations” (guesses) to find the nonce, requiring a total of 35 TWh (terawatt hours) per year. At $0.10 per kWh, this means it cost approximately $3.5B to validate the bitcoin blockchain in 2017, although wholesale electricity costs are be considerably lower in certain regions. The environmental implications of this waste are the

Solutions based on PoS blockchains such as Hyperledger fabric have a different set of problems. It is true that using a relatively small number of stake-holders decreases the computational (and other) costs of validating transactions. If only ten or twenty stake-holders are to be trusted with the job of validating millions or billions of dollars in transactions, however, they must be carefully screened. As a result, their identities are known and users must ultimately trust that the screening process is effective and will remain so. This runs completely contrary to the underlying idea of blockchain as a trustless, decentralized, and distributed ledger technology.

Some approaches to PoS involve larger numbers of stakeholders, but require that they choose to give their share of the voting power to a smaller number of delegates. This speeds transaction confirmation since a relatively small number of nodes/stakeholders are actually doing the work of verification. Such systems are only secure, however, to the extent that the bonds posted by the stakeholders are large in comparison to what they could gain by acting dishonestly. Since posting bonds is costly, these PoS approaches also bake in an inescapable level of transactions costs (similar to those discussed for PoW protocols).

In short, there is no such thing as a free lunch. Either transactions costs are high, or validation is in the hands of a small number of possibly nonanonymous agents. Unless a solution can be found, cryptocurrencies will never be secure enough to offer a serious challenge to the conventional banking system.

2.3 Scalability

Both the Ethereum and Bitcoin blockchains are operating near maximum capacity. The Ethereum blockchain writes blocks about every 12 seconds and currently processes between 4 and 7 transactions per second (with an estimated maximum rate of 15 per second). Bitcoin writes blocks every 10 minutes and processes 2 to 4 transactions per second (with an estimated maximum rate of 7 per second). Neither protocol would be able to scale up to the 2000 transactions per second handled by the Visa network, which has an estimated maximum rate of 56,000 per second.

Bitcoin's proposed solution to this problem is called the lightning network and is similar to Ethereum's raiden network. Essentially, users are required to lockup tokens on the main Bitcoin or Ethereum blockchains to serve as security for transactions that agents agree to off of the main-chain. These off-chain transactions are not validated or committed by the mining pool, but there is a degree of security provided by a system of smart contracts. This allows both parties to cancel or alter transactions until they mutually agree that a transaction is final. There are a great many problems with this approach, but we will not go into detail here.[12] However, it is worth pointing out that users must pay normal transactions fees to move coins onto the Ethereum or Bitcoin main-chains and lock them into escrow in order to make them available for use on the raiden/lightning networks. Fees also must be paid to bring the results of activities on these side networks back to the main-chains. In a sense, these networks allow users to place value on debit cards that they can use to

same regardless of the cost of electricity.

[12] For example, see John Ratcliff's analysis of the lightning network at: http://codesuppository.blogspot.com/2016/02/the-lightning-network-is-so-great-that.html

execute transactions quickly and cheaply without involving the “bank” (the main-chains in this case). However, this requires that the “bank” both issue the cards and redeem them, in order to return whatever value they contain back to the users’ main-chain account.

As discussed in the previous section, PoS based solutions either use a small number of validators or run into the same scaling issues as PoW approaches.

Iota’s tangle protocol is another approach to increasing transaction capacity. It is not actually based on blockchain, but instead relies on repeated validation of transactions by individual nodes. Eventually, users are supposed to gain confidence that a transaction is valid and finalized based on an evaluation of the number of independent confirmations and the reputation of the nodes involved. The BFT of tangle is not clear, and it appears to be open to strategic manipulation on a number of fronts.[13]

Micropayments are only one use case where high transaction volumes are likely. For example, stock exchanges and other financial markets could only be moved to blockchains if they had the capacity to handle hundreds or thousands of transactions per second. There exist billions of connected devices today ranging from medical and industrial sensors to household thermostats and appliances. With autonomous vehicles just over the horizon, the Internet of Things (IoT) is poised to become even more important. Creating networks that allow such devices to safely interact as agents for their owners, or to create and share verifiable records is only possible if blockchain protocols exist that can record large volumes of low value transactions. Even keeping ordinary records for hospitals, government agencies, or large companies might require writing many hundreds of data items into a blockchain per second.

3. GeeqChain Basics

GeeqChain is a flexible platform that is built around interoperable federated instances of blockchain tailored to different use cases. Basic protocols and the custom business logic of GeeqChain instances is encoded in a series of Governing Smart Contracts (GSCs), which provide the components of the user and node clients (the software used to run and interact with GeeqChain). These GSCs are included in genesis blocks that are the foundation of each instance. GeeqChain also takes a novel approach to network topology and communications. When these are combined with a type of unanimity game to achieve consensus, the result is a validation protocol that offers Strategically Provable Security (SPS),[14] effectively, 100% BFT.

3.1 Genesis Blocks

All instances of GeeqChain begin with a genesis block (block number 1). Blocks of validated transactions are created by nodes in the network and appended sequentially. Genesis blocks are

[13] For example, see Christine Masters’ discussion at: https://cryptovest.com/education/not-a-iota-the-trouble-with-iota-and-how-to-fix-it/.

[14] For a discussion with much greater detail, see Section 4.

created by GeeqCorp at the request of other companies who wish to build applications. This tight control has several motivations.

- It prevents duplication of chain, token, and controlling authority names.
- It ensures that the chain adheres to the CDM protocol for transaction verification. The SPS that results makes it possible for federated GeeqChains to trust in the integrity of other ledgers and to accept tokens from other GeeqChains.
- It fixes the rules under which the chain will operate. This is done by including copies of the GSCs in the genesis block. GeeqChains can be adapted to many purposes, using different sorts of business logic for native tokens. While the rules for what make a valid GeeqCoin transaction are universal to all federated chains, native tokens might be used to distribute votes, governing power, or profits, as rewards for users (as distinct from validating nodes), to tokenize real assets such as stocks, bonds and land titles, and so on. Users can verify that the rules are being followed by doing their own audits and verifications with the help of the GSCs written into the genesis block. Note that the business logic these GSCs contain may also rely on data records written by users into the chain in addition to token transactions.
- The genesis block also contains pre-mined tokens and/or sets out the rules under which tokens can be created in the future. Users know going in exactly how the token economics for any given instance of GeeqChain will work as a result.
- The business model of GeeqCorp involves payment of GeeqCoins to GeeqCorp accounts in various ways. These may include periodic fixed license fees, transactions fees, or small fees to update the ledger states, for example. These payments are automatic and built into the GSCs with the agreement of a client who wishes to start an instance of GeeqChain.

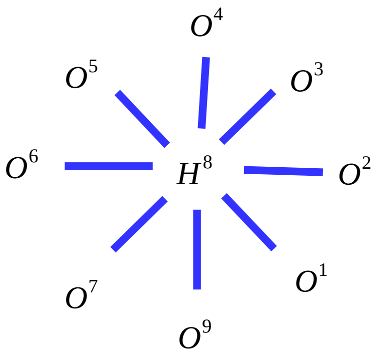
This might seem to give GeeqCorp a great deal of power that could lead to potential abuses. However, once a genesis block is created, neither GeeqCorp nor anyone else can alter it or the operation of the chain that uses it as a foundation. There is simply no mechanism to allow such manipulations to happen. In addition, each instance of GeeqChain can produce only the type of tokens listed in the genesis block, and no other instance can produce the same token type. An implication of this is that only the original GeeqChain will produce GeeqCoin, and since these will all be pre-mined, no additional GeeqCoins can ever be created.

3.3 Network and Communications

Nodes are validators of transactions submitted by users for inclusion in new blocks of the GeeqChain. Real world agents (that is, people) download and install a copy of the GeeqChain node client on a computer which can be reached at some IP address. Each active node builds, keeps, and makes available to users its own copy of the GeeqChain.

The CDM uses a unanimity rule for consensus. Knowing the state of the network and who did or did not send and receive messages is a necessary condition in order to determine unanimity. However, communicating with each and every peer would use a great deal of bandwidth. If the CDM required such extensive communication to function, it would also introduce a great many points of failure.

The solution is based on two basic elements. First, the CDM uses a random hub and spoke approach. The simplest case is for a single hub to coordinate the building of each block. For example in a nine node network, if node number 8 gets randomly chosen to serve as the hub for building block number B, the network would be the following:



After the block B is built and committed, a new hub is chosen randomly to coordinate the building of block B+1. More complicated structures with multiple layers of hubs and random participation rules are possible and may be desirable, depending on the needs of the chain being validated.

Second, the CDM uses a special GSC to handle communications between nodes, clients and users.[15] In brief, this element makes it possible to prove and verify whether messages were sent and received, and by whom. More generally, the network communications GSC closes off attack surfaces that dishonest nodes might exploit, but leaves the CDM robust to even extreme network failure.

3.4 An Example of a Consensus Game

The CDM is based on game theory and economic mechanism design. To get a sense for how these disciplines can be used to produce SPS for blockchain validation, consider the following game:[16]

[15] The CAP Theorem tells us that if we insist that all nodes in a network communicate and agree on a ledger update, we cannot guarantee that consensus will be achieved in finite time. In other words, if a mechanism sets an upper bound on the time to achieve consensus on a ledger update, then it must be robust to the possibility that some nodes will either not communicate with the network or will have different or incomplete views of the ledger update.

[16] The unanimity game described is an example of an economic mechanism based on non-cooperative game theory. The objective of mechanism design is to create a structure such that when agents act purely in the their own self-interest, the result is a desirable outcome such as truth-telling, optimal matching, or revenue maximization. Second price auctions used by eBay and Google AdWords, and the National Resident Matching Program that assigns new MDs to hospitals for training in medical specialties are important and well-known examples. In a formal sense, the CDM is a

Agents are offered a chance to play a game in exchange for a one dollar admission fee. Each player who pays the fee is sent to a room where a word is written on the wall. Players are asked to write this word on a piece of paper. The papers are then gathered and compared. If they all have the same word, then each player is paid two dollars. If there is any disagreement about the word, all players get zero (which gives each a net payoff of negative one dollar).

It is easy to see that truth-telling is a Nash equilibrium.[17] Suppose that one agent sees that all other players have reported the truth. Clearly, his best response is to tell the truth as well. Reporting the correct word gives the agent a payoff of $\$1$ while any other report gives him a payoff of $-\$1$.

Unfortunately, this game has many other Nash equilibria as well. For example, suppose that at least three players make different reports. All players would get a payoff of $-\$1$ in this case. Since no single player could change his report and generate unanimity, all the player's strategy choices result in a payoff of $-\$1$. In other words, all reports are (equally bad) best responses for any individual agent. Alternatively, all the players might get together before the game is played and agree to coordinate on one specific untrue report. In that case, each player would get a payoff of $\$1$ and no single player would benefit from unilaterally changing his report and telling the truth. Of course, coordinating on a false report does not yield a larger payoff than coordinating on the truth, nevertheless, both are Nash equilibria.

In the context of blockchain, it might be the case that players could profit substantially if they coordinated their efforts. Suppose we modified the game to assume that a name instead of a word was written on the wall, and that if all players write down the same name, then the named person would get $\$1000$. Truth-telling and discoordinated reports would still be Nash equilibria, however, players would profit more if they coordinated. For example, they could agree to write down one of their own names and then split the $\$1000$ received. In so doing, players would still get the $\$1$ payoff for their unanimous reports but would get an equal share of $\$1000$ in addition. Agents, therefore, have a positive incentive to collude, unlike the simple game first described.

To fix this, we could alter the game again to allow a small amount of auditing. Suppose we required all agents to sign their reports. If the reports are unanimous, then agents get a payoff of$\$1$ and $\$1000$ goes to the person they name. However, if the reports are not unanimous, then the door to the room is opened, and the name on the wall is read. Any player who wrote down the correct name would get a payoff of $\$1$ plus an equal share of a $\$1000$ bonus. Players who lie would receive nothing and be banned from ever playing the game again.

With this addition, truth-telling becomes a dominant strategy. That is, regardless of what other players report, it is a best response for each individual is to report the truth. Better still, truth-telling is a coalition-proof equilibrium. That is, no group or coalition of agents, including the coalition containing all the agents, could profit from lying. Even if all agents were able to agree on a false report,

mechanism that implements truthful validation of blockchains in a coalition-proof equilibrium.

[17] The Nash equilibrium concept is based on the idea that all agents unilaterally choose a "best response", assuming a fixed set of strategy choices on the part of all the other players.

any single agent who reneged on the agreement and told the truth instead would get a payoff of $1001. If several agents reneged, they would get to share the $1000 bonus divided among the set of defectors (which would be more than $1000 shared over all players that a coordinated false report gives). Thus, truth-telling is always a better strategy than lying or trying to collude. In other words, truth-telling is the *only* coalition-proof equilibrium. As a result, there would never actually be a need to do an audit and pay the $1000 bonus.

4. GeeqChain as a Solution

GeeqChain has the benefit of learning from Ethereum, which in turn, had the benefit of learning from Bitcoin. Just as Ethereum solved many of the limitations embedded in the Bitcoin protocol, GeeqChain solves most of problems remaining in Ethereum. In particular, GeeqChain is secure, cheap, fast, and scalable. It can be implemented with fully anonymous verifying nodes, no centralized points of trust or failure, and any level of encryption and privacy protection desired.

4.1 Security

In most blockchain protocols including Bitcoin, Ethereum, and GeeqChain, network latency, failures, and attacks can create communications delays and even cut parts of the network off from one another. This can result in a “fork” in which two different blocks are built on top of the same initial chain by different sets of nodes in the network. That is, while all nodes may hold identical copies of the chain up to block number B, subsets of nodes may end up adding a B+1 block that contain differing sets of transactions. Having two conflicting versions of a blockchain ledger is untenable. Something has to be done to choose which one is definitive or else users might try to spend coins on each of the chains. Since users’ accounts exist on both forks, such double spending would not be detected as fraudulent.

The traditional solution used by Bitcoin and Ethereum requires that the next miner who completes a block adds it to the longest fork of the chain. If both forks are of equal size, the miner is allowed to choose either one. Eventually, one fork gets ahead, and the shorter fork is “orphaned”. All the blocks that were added after the forks diverged are ignored along with the transactions they contain. The choice of which fork to use is ultimately up to the miners who find new blocks, but protocol requires that they follow these rules.

GeeqChain, using CDM, takes a completely different approach to validation called Proof of Honesty (PoH). Each node validates transactions, builds blocks, and publishes them for users to inspect. These blocks contain enough data for users to independently verify that the chain is “honest” in the sense that it contains only valid transactions and that the chain as whole follows protocol. In other words, nodes, and the chains they construct, are provably honest or dishonest.

If one or more forks exist, the user is able to inspect them and then decide which one he wishes to use for his transactions. The CDM places the responsibility of choosing between forks in the hands of users who hold tokens on the chain and who would therefore be harmed if false transactions were written into "verified" blocks.[18] Thus, Bitcoin and Ethereum have an inherent conflict of interest built into their protocols since nodes, who may benefit from writing false transactions, choose which fork to add newly mined blocks to. GeeqChain's CDM protocol, on the other hand, makes this choice incentive compatible by leaving it to users.

Without further elaboration, this simple idea gives the GeeqChain a BFT of 99%. That is, if even a single honest node exists, then users will discover it and choose to write their transactions to the chain it verifies. While this is far better than any other protocol, GeeqChain aims higher.

The CDM builds on the intuition of the unanimity game described in the previous section, but turns it on its head. Specifically, the CDM achieves consensus by checking for a *lack of dissent* rather than attempting to affirmatively establish unanimous agreement that a block is valid. If a user or node detects any dishonesty, he can send an audit request to the network. If dishonesty is verified, misbehaving nodes are ejected from the validation network and Good Behavior Bonds (GBBs), previously posted in order to join the network, are forfeited and used to pay rewards to the agents who called for the audit. These audits and rewards combined with the certainty that dishonest behavior will be detected (due to the network protocol and the design of GSCs) create an incentive structure that makes it impossible for agents, even working in coordinated coalitions, to profit from dishonest behavior. As a result, truthful validation by all nodes is a coalition-proof equilibrium (in fact, the only one). In effect, GeeqChain is 100% BFT and offers users Strategically Provable Security (SPS) for blockchain validation.

4.2 Cost

Consider a GeeqChain processing 40 transactions per second and writing blocks every 10 seconds. This is a higher transactions volume than either Bitcoin or Ethereum are capable of. Further suppose that these transactions are validated on a network consisting of 100 nodes. This is far fewer than the 32,000 or so nodes on the Ethereum network or the 10,000 or so Bitcoin nodes. On the other hand, it is greater than the 25 or so nodes that validate some implementations of the Hyperledger fabric protocol and other PoS consensus systems. Finally, suppose that an average transaction contains 0.5kB of data, roughly in line with Ethereum and Bitcoin transactions.

Using high-side estimates of the amount of bandwidth, compute cycles, storage, and of the costs of these resources, we find that the total cost validation on the network outlined above to be less than $\$.0006$ per transaction. As an example, a transaction of $1¢$ could be validated, committed, and stored on this CDM blockchain for less than $.06¢$. In contrast, Ethereum transactions cost on the scale of $15¢$ or more and Bitcoin transactions fees can run to several dollars. The cost per transaction scales linearly in the number of nodes. As a result a GeeqChain that uses N nodes in its validation network, can process a transaction for less than

$$\$6N\times10^{-6}.$$

[18] Checking the honesty of fork is done automatically by the user's client software provided by GeeqChain, and so this responsibility in effortless and invisible from the user's perspective. However, users may choose do due diligence on their own or use other means to decide whether a chain is honest if they wish.

We explore the implications of this fact in more detail below.

4.3 Scalability

One of the most serious limitations of existing blockchain protocols is scalability. Estimates are that the maximum number of transactions per second is 7 for Bitcoin, 15 for Ethereum, 60 for Dash, 30 for Monero, and 20 for Litecoin. Average block completion time ranges from 10 seconds to 10 minutes. Even then, transactions are not considered finalized until they are buried several blocks deep in the chain (six or more for Bitcoin and 250 or so for Ethereum).[19] Even if user transactions are immediately picked up by a validating node and included in the next block, it still takes an hour or more for the transaction to be considered final.[20]

Consider the GeeqChain instance described above (100 nodes and 40 transactions per second). The first question to answer is whether it would be feasible to run a GeeqChain with this configuration on standard home computers and broadband connections. The random hub and spoke network architecture implies that each validating node has an equal chance of being the hub for any given block. Thus, each node serves as hub 1% of the time. We calculate that a hub would need to upload 20MB of data in the 10 seconds allocated to build a block. This means that a hub would need to upload data at a rate of 2MB/s or 16Mb/s. According to speedtest.net, the average upload speed for US residential broadband customers in 2017 is 22.69Mb/s. This means that the average US household has a fast enough internet connection to run a hub on the network just described. For the other 99% of the time, verifiers act as ordinary nodes and receive only 200kB (.0002GB) per 10 seconds, requiring .16 Mb/s of download speed.[21]

More generally, the amount of data that a hub must upload scales linearly with the number of nodes, the number of transactions per second, and the size of transactions. The download capacity for simple nodes scales linearly with the number and size of transactions. Home broadband connections have more than enough capacity to run an ordinary node, but hubs may present a problem. To see this, suppose there were 1000 nodes on the network and 100 transactions per second. Then hubs would need upload rates of 400Mb/s, more than most home broadband connections offer. One solution would be to run nodes and hubs on virtual machines on a cloud platform that could make this level of bandwidth available when needed. The GeeqChain network could also be structured as a layered system of random hubs which would make it feasible to use thousands of nodes without exceeding the capacity of residential broadband connections.

The computational load of running a GeeqChain node (or hub) is very small. Home computers should easily be able to handle networks of arbitrary size running at 40 or more transactions per

[19] https://www.cypherium.io/index.php/2017/03/26/why-scalability-matters/

[20] Gou (2017) notes that "as of June 2017, there are more than a quarter million unconfirmed transactions within the Bitcoin network mempool. Executing a single Bitcoin transaction can take between a few hours to a few days, with transaction fees averaging several dollars and rising with the increasing demand of the block-size market."

[21] Recall that 1GB = 1000MB =1,000,000kB. Thus, 200kB = .2MB = .0002GB. Broadband speeds are specified in megabits per second (Mb/s). Since 1 Byte (B) equals 8 bits (b), 200kB = 1.6 Mb which in turn requires a connection speed of .16 Mb/s to be transmitted in 10 seconds.

second. The efficiency of GeeqChain's network allows it to handle a larger transactions load than either Bitcoin or Ethereum. Ultimately, the number of transactions per second is only limited by the upload bandwidth available to nodes.

Bitcoin, Ethereum, and most other blockchains trade only in their own native tokens. In particular, it is not possible to move a bitcoin to the Ethereum blockchain or inversely. There is only a single master chain for each of these tokens where transactions can be made and token holdings recorded.

GeeqChain, in contrast, is designed to support multiple instances of federated chains that form an ecosystem in which users can choose where their tokens are parked. This federated structure gives GeeqChain a flexibility that allows it to be adapted to many use cases, some of which we discuss in the next section. If 40 transactions per second (more than either Bitcoin or Ethereum can handle) is not enough, new federated instances of GeeqChain can be created by dividing the set of nodes and accounts on the original chain in two. These instances would share the job of validating transactions, and tokens would be able to move freely between them. Since any number of instances can be created, GeeqChain can be scaled up to handle arbitrarily large transactions loads.

Moving tokens from one chain to another requires that they be destroyed on the sending chain before they are created on the receiving chain. A good way to think about moving tokens between chains is that tokens representing assets can be "teleported" from chain to chain. The token disappears on one chain and reappears on another. Protocols are designed to prevent "teleportation accidents" where the token is duplicated elsewhere while not being destroyed on the originating chain, is sent to more than one receiving chain, or is somehow held in a pattern buffer and then recreated later on the originating chain. Of course, it is also desirable to prevent accidents in which tokens fail to materialize on the destination chain, but the consequences of this are far less damaging than unintended duplication.[22] On the other hand, non-token data items such as medical records could be "replicated" locally on several different chains without harm. The owners or creators of these replicated items might be paid fees for allowing this.

4.4 Updating Protocols without Violating Protocol

Blockchain protocols are inflexible by design. Allowing trustless interactions between anonymous agents requires that the rules be well understood and unchanging (Code is Law). The downside is that useful upgrades and fixes can only be made through hard forks or complicated governance systems.[23]

[22] Recall the consequences of having two Will Rikers in *Star Trek: The Next Generation*, season 6, episode 24 (Second Chances). Unauthorized replication of tokens would completely undermine both the token economics and user confidence in the underlying blockchain. The loss of a star fleet officer or token, on the other hand, is unfortunate, but the risk is more at the level of an individual misfortune than an existential threat to the system.

[23] For example, on November 9, 2017, bitcoin decided against a soft fork adding SegWit technology that would have improved performance by moving some unessential data off the underlying blockchain. On the other hand, Bitcoin Cash was implemented as a hard fork in August of 2017 in order to increase block size and allow its blockchain to handle eight times as much data as the original bitcoin. The introduction of Bitcoin Cash through a hard fork resulted in a steep and immediate decline in bitcoin price and a sharp increase in value of the Bitcoin Cash token. Clearly,

The system of federated chains that the CDM permits provides an elegant solution to this problem. If an instance of a GeeqChain is found to have bugs, has become obsolete, is not taking advantage of new technologies or meeting current user demands, a new instance could be created with the intention that it replace the existing instance. The new chain would have a genesis block with a new set of validators and GSCs, but no tokens. Users and validators on the old chain could choose to move to the new chain where the rules are different or they could stay where they are. If they choose to move their tokens, their actions are voluntary and within protocol. If they choose to stay, they can continue to live under the old rules. If enough users and nodes support the continued existence of the original chain, they can trade tokens under the original rules indefinitely.

It will not be too long before 256 bit (or greater) encryption can be broken by quantum computers. This will undermine the security models of all existing blockchains and almost everything else in the cloud. Fortunately, with quantum computational approaches to breaking encryption will come new quantum-proof approaches to encrypting data. GeeqChain's federated architecture allows the creation of new quantum-ready instances of existing chains and applications for users to migrate to as quantum technology matures. This makes GeeqChain more future-proof than any existing blockchain platform.

5. Applications

In this section, we outline a number of possible applications that could be built on GeeqChain genesis blocks.

5.1 Micropayment Platform

The low cost and high transactions volume of GeeqChain make it ideally suited for use as a micropayments platform. This might be implemented as part of a smart city system to allow citizens to pay for parking, bridge tolls, subway fares, items in vending machines, or minor city services. Consumers could use an instance of GeeqChain to buy entertainment, gaming, and other content on the Internet from various providers. IoT devices could make micropayments via GeeqChain to buy and sell services (CPU cycles, sharded storage, or electricity produced by solar panels, for example).

As an example of the revenue potential of GeeqChain, consider a micropayment platform using the following fee structure (only one of many possible fee structures):

Fixed fee to the node receiving the transaction:	.1¢
Percentage fee paid and divided over nodes:	.25%
Percentage fee paid to GeeqCorp:	.25%
Maximum fee for any transaction:	10.1¢

Assume the following level of use:

there is demand for a mechanism to upgrade and change blockchain protocols, but the current necessity of using forks to do so makes it costly and contentious, and introduces tremendous unpredictability in markets.

Transactions per second:	10
Average transaction amount:	25¢
No transactions over $20	

If there are 10 transactions per second, there would be a total of approximately 300M per year with a total value of about $80M. If there were 100 validating nodes, each node would receive about 3M transactions per year and be paid $3k in fixed fees. The nodes collectively, and GeeqCorp individually, each get a fee of .25% of the total transactions value, or $200k. This gives each node a revenue of $5k and a net profit of $3.2k. GeeqCorp gets a pure profit of $200k.

5.2 General Payment Platform

Suppose instead that an instance of GeeqChain was deployed as a general payment platform such as PayPal or Visa/MC. Using the same very low fee structure, suppose there are 100 validating nodes, 10 transactions per second, but all transactions are over $20. This gives an annual transaction volume of over $60B. Given that Bitcoin volume in 2017 is more than $300B as of November 15, this is not an unreasonably high estimate. In this example, nodes get a revenue of $150k per year while GeeqCorp gets a net profit of $15M.

5.3 University Student ID Card Cashless Payments

Thousands of universities, hospitals, and corporations with large campuses use third parties to enable cashless payment systems using ID cards. These cards may be restricted to meal plans and internal fees, extend to dollar transactions at bookstores and local businesses, and may even incorporate credit and debit card functions. CBORD is one of the major providers of such services and charges fees to participating merchants of up to 6%. Additional fees are charged for ATM use, debit card transactions, and even for recharging prepaid versions of such cards. Hong Kong has long used the “Octopus Card” for bus and subway fares and now has an extensive network of merchants who accept this prepaid card for various goods and services. Other cards of this type are linked directly to the VISA or MC payment networks and have standard credit card fees.

A university or corporate environment offers two significant advantages that make it ideal for GeeqChain payment systems. First, all the users are known to the system and pre-vetted. This means that KYC and AML[24] compliance is comparatively easy. Second, the university or corporation stands as the guarantor of the tokenized dollars that move over the network.[25] In the simplest

[24] KYC (Know Your Customer) and AML (Anti-Money Laundering) rules are imposed on banks and other financial institutions to prevent tax evasion, criminal activity, and the funding of terrorist activities. Banks, for example, are required to check identification documents of customers and report earnings and large movements of money to the federal government. Cryptocurrencies serve the same purpose as fiat currencies and create the same potential for illegal activities. The SEC and other agencies are becoming increasingly insistent that blockchain platforms also comply with these rules.

[25] Of course, a university or business could default on these obligations, but then there would be legal recourse available against an established entity with a physical presence. Universities and corporations also have a strong reputational incentive not to default if they wish to maintain good relations with students, alumni, employees, suppliers, and customers. In contrast, Tether (https://tether.to/) is a blockchain platform offering tokenized dollars that it claims are backed 100% by dollars on deposit in banks in Shanghai and Taipei. More than $1.3B worth of Tether tokens exist

case, students pay dollars into student accounts, and the university creates a corresponding number of tokens in the student's account on the university's GeeqChain. Merchants and others who accept these tokens from students through the campus GeeqChain can request that the tokens be transferred by the university to their own bank accounts via a free ACH transaction. The tokens would then be removed from the chain.

Closed payment systems such as these would require little if any technical expertise on the part of the token authority and the transactions costs of keeping accounts on the chain would be extremely small. The only potentially significant cost would be establishing a reliable Point of Sale authentication system to allow the cards to be used. Cellphone apps with multifactor identification is one possible approach to solving this problem.

5.4 Tokenized Trading of Assets

Assets such as stocks, bonds, real estate, and commodities such as gold, can be represented by tokens on a blockchain, either through a simple system of accounts, or a system in which each individual item is considered a separate, unique object to be traded. In the latter case, the business logic of the host GeeqChain would need to be customized to require that each unique token exist in one and only one account to be considered valid. Another option might be a mixed approach where stocks issued by a given company are represented as homogeneous tokens that use the standard Bitcoin-type accounting logic. Stocks issued by different companies, or different types of stock issued by the same company (preferred or common, for example), would use different tokens that may not be added together. Thus, an agent might have 50 shares of IBM and 75 shares of Microsoft. IBM shares would be represented by homogeneous tokens but of a type that could not be added together with similarly homogeneous Microsoft tokens.

Tokenized exchanges would be built on federated instances of GeeqChain and transactions fees paid to validating nodes in GeeqCoin. Since every instance has Strategically Provable Security for validation, all asset tokens and GeeqCoin transactions fees may be safely moved between accounts of different types on different chains. A company that wished to make a stock issue would obtain a genesis block signed by GeeqCorp containing colored tokens to represent the asset. The genesis block would contain no GeeqCoins, so paying transactions fees would require that they be moved onto the new chain from another federated GeeqChain. From then on, the new chain would be able to interact with the rest of the GeeqChain ecosystem.

It might be convenient to maintain each asset/token type on a separate chain (for example, all IBM stock might trade on its own instance of GeeqChain). Alternatively, brokerage houses might set up separate instances of a GeeqChain for their clients, each of which holds many different types of tokens. These in-house chains might include logic to write balances to accounts with encrypted information available only to the brokerage house to allow it to identify the clients behind the public

as of the end of 2017. Although there is no technical reason this should not be an excellent way to provide a stablecoin on a blockchain, several problems have arisen. AML and KYC have proven to be difficult and the international banking system has all but cut off relations with Tether. In addition, Tether has not been completely transparent in allowing audits to show that all the needed dollars are in fact on deposit. There is no real legal guarantee that they would stay on deposit in any event.

key address. This would facilitate KYC and AML efforts, reporting information to the proper tax authorities, and allow the brokerage house to create aggregated reports for their clients. All of these functions are possible using federated chains, and all would be done without the brokerage house assuming actual control of a client's assets. Provided the private key remained with the user, only the user would be able to write a valid transaction. As a result, embezzling from clients becomes impossible, while client privacy is protected to the extent that the law allows.

A similar approach would work for registering physical assets such as cars or property titles. Since each object is unique, it would be represented by a uniquely identified GeeqToken. Federated chains holding such tokens would include logic that allows an organizational unit, such as a state department of motor vehicles or county clerk (called a "token authority"), to create signed tokens attesting to the car's existence or the validity of the deed. Reassigning ownership or removing the registration of an asset would be a simple, uniform, streamlined and low cost process using the token authority's GeeqChain. Different token authorities could also cooperate and coordinate using their federated chains to transfer auto titles to different states or to help other agencies collect property taxes, for example.

GeeqCorp would charge fees for issuing genesis blocks, licensing fees for use of the IP, or could include business logic on the new chains that give GeeqCorp a fee for various transactions or executions of protocols.

5.5 Validation of Records and IoT data

Public records, transaction audit trails, the activities of medical devices, and telemetry for sensors and alarms, are all examples of cases where there are large volumes of data that need to be publicly available, auditable, shareable, and/or provable, but where each data item individually has low value. The fact that GeeqChain provides SPS quality validation and time stamp services at a marginal cost of $.0006 per item (or even less) makes it economically practical to place such data on blockchains. Revenue models might be as simple as charging a multiple of the marginal cost per item, or licensing the use of an instance of GeeqChain to a municipality, hospital chain, or IoT manufacturer. Logistics chains, two-sided markets using blockchain, and cases with more valuable data, would use similar revenue models.

5.6 Generalized Platform for Other Blockchain Startups:

Ethereum has created an ecosystem of blockchain startups through its ERC20 token standard. Startups create their own tokens in a fairly flexible way and then use an Ethereum smart contract to validate transactions. ERC20 contracts contain a mapping of public key addresses to balances of native tokens (the ledger state of the ERC20 token) and rules for moving tokens between accounts. The ledger state in the contract is updated as users make transactions and, as a result, startups can take advantage of Ethereum's mining network for validation.

This all comes at a cost. Updating the ledger-state in the ERC20 contract requires paying the Ethereum Virtual Machine (EVM) to run the contract each time. If the business logic used by the startup's contract requires using off-chain data from oracles, users, or other smart contracts, EVM usage becomes more intense. Even for very simple transactions, average Ethereum fees are around $.15. Keeping the ledger state current in more complex ERC20 contracts could turn out to be quite expensive.

One solution is to keep transactions, data, and the ledger state in a private side-chain created by the startup. Hashes of the ledger state might be written to the Ethereum chain periodically to provide PoW validated checkpoints that could be used to verify the continuity of the history of the side-chain. It is usually impractical to have the side-chain itself validated by PoW since that would require establishing and paying a large network of dedicated miners. As a result, PoS is used in most cases. We have already discussed the security problems with PoS which multiply when the set of stake-holders is small or poorly compensated, as they usually are on new side-chains. Thus, side-chains offer limited security for users and are difficult and potentially expensive to setup.

Another alternative is to keep the data needed to support the business logic of a platform (and maybe even transactions waiting to be written to the ERC20 contract) in a private database. For example, a logistics chain might keep signed receipts handing off cargoes between truckers, warehouse managers, shippers, and other users in a database and give access only to those who have a need to know. Reputational data regarding agents, escrowed tokens, or bids and asks in two-sided markets might similarly be kept off-chain in databases, which save the costs of writing each item or transaction to a verified Ethereum block. The problem with this is that it requires users to trust the platform that holds the data. What if the platform's servers are not secure? What prevents authorized agents or even the trusted platform keeping the data from writing false entries? In any event, if users can trust the platform, why is there a need for a blockchain to execute and record the ERC20 token or any other type of payment? After all, if the underlying data is corrupted, then the payments will be wrong, and if the data is correct, then users may as well trust the platform to credit accounts honestly.

GeeqChain provides an alternative to ERC20. New instances of federated chains can be created through the issue of a signed genesis block from GeeqCorp. Transactions involving GeeqCoins on these new chains would follow standard GeeqCoin protocol. Native tokens, however, could follow their own business logic as needed by the application. Smart contracts would be run on the new chain itself and would not depend on any central chain for validation. The main advantages of such an approach are that it is cheaper, faster, and also more secure. Transactions would cost fractions of a cent instead of fractions of a dollar. Transactions would also be finalized as blocks were written and would be picked up for validation as they were submitted. Finally, all data relevant to the business logic of the application would be written immutably into the platform's genesis block and verified by a protocol offering SPS.

6. Conclusion

GeeqChain is a scalable, inexpensive, and computationally light approach to validating blockchains using a new protocol called the Catastrophic Dissent Mechanism (CDM). The CDM uses anonymous actors as validators who are free to join and leave the system as they please. The mechanism gives all actors strong incentives to behave honestly, both as individuals, and as members of coalitions who might benefit from compromising the integrity of the blockchain. If there is at least one honest node, it will write an honest block to a valid chain. Users are able to discover honest chains and will always choose it for their transactions. Dishonest chains become orphaned. In other words, GeeqChains protect the integrity of blocks and transactions using Proof of Honesty rather than Proof of Work or Proof of Stake. If even one node is honest, no tokens can be stolen from conscientious users. As a result, GeeqChain is 99% Byzantine Fault Tolerant (BFT) and achieves consensus by checking for lack of dissent, rather than by affirmatively trying to establish unanimity.

We further develop a communications protocol and a system of self-enforcing audits that imply that honest behavior on the part of all nodes is the only coalition-proof equilibrium of the validation mechanism. That is, even if nodes are free to communicate, collude, and conspire to act in unison, any self-interested coalition will find that honest behavior gives its members the highest possible payoff. As a result, the CDM creates a blockchain with Strategically Provable Security.

The CDM is a flexible protocol which allows new instances of GeeqChains to be created and then specialized in a number of ways. In particular, it is easy to construct federated instances of GeeqChains such that tokens can move across them all. This brings three key advantages to GeeqChain. First, it is possible to split a GeeqChain into two or more federated instances and partition the user accounts among them. As a result, GeeqChain is infinitely scalable. If the transactions load becomes too large for one chain to handle, new federated instances can be created until each handles an efficient number of transactions per second. Federated instances can also be merged if transactions volume drops off. Second, new instances of federated chains that use business logic or protocols that are different from the basic CDM outlined here, or which contain upgrades or bug fixes, can be created. Users and validators can choose to remain on the original chain or migrate to the new one. As a result, improvements, alterations, and fixes can be implemented without breaking protocol or creating hard forks. In addition, this does not require a complicated governance structure to implement. Users and validators can vote with their feet if they agree it would be better to use the new chain than continue to use the original one. Finally, federated instances of GeeqChain can also work with tokens besides GeeqCoin, using many different types of business logic while offering Strategically Provable Security. As a result, GeeqChain offers a flexible alternative to Ethereum's ERC20 standard on which to build new blockchain platforms.

Perhaps the most import aspect of GeeqChain is that it offers this level of security and flexibility at extremely low cost. It is not burdened by large networks doing proof of work or stake-holders that need to be compensated for posting large bonds. Validating networks of arbitrary size can be used, in which each node can still be run on a standard home computer using existing broadband connections. The cost of validating transactions on a 100 node network is less than .06¢ and scales up linearly with the number of nodes. This creates large potential for revenue in every application or and opportunities to provide services on a large scale at very low cost.

In conclusion, the GeeqChain platform using the CDM protocol solves the most significant outstanding problems facing Blockchain today. GeeqChain can scale to handle arbitrary numbers of transactions per second. GeeqChain can be deployed as a system of federated chains that share a common token or which interact with heterogeneous tokens using different business logic. GeeqChains can be upgraded or altered without the use of hard forks or breaking the rule that code is law. GeeqChain can be implemented with enough anonymity and decentralization to protect user privacy and satisfy most cryptoanarchists, or to comply with KYC, AML and other regulatory requirements. Finally, GeeqChain offers an unprecedented level of transaction verification security at a lower cost than any existing platform.

References

Cachin, C. and M. Vukolic (2017) “Blockchain Consensus Protocols in the Wild” ArXiv e-prints, http://adsabs.harvard.edu/abs/2017arXiv170701873C

Eyal, I and E. G. Sirer (2014) “Majority is not enough: Bitcoin mining is vulnerable” In Financial Cryptography and Data Security 18th International Conference, FC 2014, pages 436–454

Gou, S, (2017) “Cypherium: A Scalable and Permissionless Smart Contract Platform”, Draft v1.0. *www.cypherium.io/wp-content/uploads/2017/03/cypherium_whitepaper.pdf*

Zheng, Z., S. Xie, H.-N. Dai, and H. Wang (2017). “Blockchain Challenges and Opportunities: A Survey” International Journal of Electric and Hybrid Vehicles, pp. 1–23.