BitCongress - Process For Blockchain Voting & Law

Written by: Morgan Rockwell
Metaballo@gmail.com

Definitions

BitCongress decentralized legislation & voting blockchain platform
AXIOMITY decentralized application & wallet for BitCongress
VOTE a Counterparty asset designated as a Vote token given to each voter
Proof-Of-Tally a tally token that is sent to each voter every time they vote
Bitcoin the world's first decentralized, peer to peer cryptocurrency & main Blockchain system
Blockchain a decentralized, peer to peer, open source, public asset ledger
Counterparty a decentralized asset creation system & decentralized asset exchange
XCP Counterparty cryptocurrency & asset
Smart Asset a tradable token created on a blockchain
Cryptocurrency the name given to a token created on a blockchain with a assumed limited supply
Smart Contract a programmable contract held in a decentralized blockchain cloud
Legislation a set of defined terms, rules and expectations for a group of people or body of power
Election a smart contract managing votes, containing rules, addresses for legislation or candidates
Voter a holder of a VOTE token created on Counterparty & attached to one's Bitcoin address
Address a cryptographic public key to accept a cryptocurrency or crypto token
Borda Count The Borda count is a single winner election method in which voters rank options

Abstract

A purely peer to peer version of electronic vote would allow online votes to be sent directly from one party to another without going through a central voting register. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent dead voters somehow voting or in digital terms, double voting. We propose a solution to the double voting problem using a peer to peer network. The network timestamps transactions, funding, candidates, comments, legislation and elections by hashing them into an ongoing chain of hash based proof of work, proof of stake, proof of tally forming a record that cannot be changed without redoing the proof chain. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of computational power which is now termed as “mining”. As long as a majority of computational power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages & data are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof of work, proof of stake, proof of tally chains as proof of what happened while they were gone. BitCongress is a combination of old world congressional concepts, blockchain technology and most importantly Bitcoin, the underlying backbone of BitCongress. A governance system built on Bitcoin, Counterparty & Smart Contracts in a separation of computational power structure.

1. Introduction

With the creation of Bitcoin & the Blockchain there has been a true demonstration of a consensus based monetary system fully functioning globally for over 8 years now. The advantages of this system shows how a peer to peer system running a decentralized node network can become the most powerful computer network on the planet in under a decade. We propose a voting system to be created in conjunction with Bitcoin, Counterparty & Smart Contracts such as RootStock.io, BitSwitch or any other Bitcoin based C++ Contract system using a distributed model to verify elections, votes and voters on separated blockchain networks. Using Bitcoin for its proof of work blockchain which has grown into the largest & safest blockchain in the world, votes will be created on the counterparty system which sits on top of Bitcoin. This will allow every vote to be hashed into the Bitcoin Blockchain, timestamped and registered on the public ledger forever. Smart Contracts will be used to create elections as smart contracts that have a set of rules to follow including election time, candidates, legislation & custom election rules. We have created a tool for legislation similar to a Bitcoin wallet or the Smart Contract Contract Maker for legislation, amendment, debate & voting called Axiomity. It will be distributed to every available market and freely accessible to anyone online. With a combination of the Bitcoin, Counterparty & Smart Contract Blockchain networks we have created a robust system to upgrade Voting, Legislation, Elections & Public Debate. BitCongress is a system that combines Bitcoin, Counterparty, Smart Contracts & a new user interface called AXIOMITY into a fully functional Congress on the Blockchain. BitCongress is a great tool for Governance, a remote control right in every phone, tablet, tv and computer, to legislate, to vote, to decide in the moment how our society should be, in an instant, on the blockchain.

2. Elections

We define an electronic vote as a chain of digital signatures. Each owner transfers the vote to the candidate or legislation by digitally signing a hash of the previous transaction and the public key of the candidate or legislation and adding these to the end of the vote. A voter can verify the signatures to verify the proof of tally. The problem of course is the voter can't verify that one of the candidates or piece of legislation ignored the vote, received the vote, faked the vote or denied the vote. A common solution is to introduce a trusted central authority, or counter, that checks every vote for identity of voter, double votes & onsite voter manipulation. After each election, the vote must be counted by a trusted authority, and only votes made directly within the centralized counter system are trusted not to be a double vote, false voter or other voter fraud. The problem with this solution is that the fate of the entire voting, election & legislation system depends on the counter counting the votes, with every election having to go through them, just like a government election. We need a way for the voter to know that the person, law or decision they voted for is recorded, counted, acknowledged & equally has power like all other votes. For our purposes, the election is held as a multi signature smart contract held between voters, candidates and legislation. This smart contract election will be running a set of rules for a set of time, able to accept votes with its public key, register them, process from them with the public key of the voter and return them to the voter after election. The only way to confirm the absence of an election, its votes & its voters is to be aware of all elections. In the central counter based model, the counter was aware of all elections and decided which votes are registered first or at all. To accomplish this without a trusted party, elections must be publicly announced, and we need a system for participants to agree on a single history of the order in which they were held, voted for and the ending results. The voter needs proof that at the time of each election, the majority of nodes agreed it was the first received.

3. Timestamp Server

The solution we propose begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash, such as in a newspaper or Usenet post. The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash. Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it. This creates a history of all elections, votes, voters, candidates, legislation, amendments, debates, quorums, filibusters & all election information.

4. Proof of Work

To implement a distributed timestamp server on a peertopeer basis, we will need to use a proof of work system similar to Adam Back's Hashcash, rather than newspaper or Usenet posts. The proof of work involves scanning for a value that when hashed, such as with SHA256 or SHA512, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash. For our timestamp network, we implement the proof of work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof of work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it. The proof of work also solves the problem of determining representation in majority decision making. If the majority were based on one IP address vote, it could be subverted by anyone able to allocate many IPs. Proof Of Work is essentially one CPU one vote. The majority decision is represented by the longest chain, which has the greatest proof of work effort invested in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. To modify a past block, an attacker would have to redo the proof of work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes. We will show later that the probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added. To compensate for increasing hardware speed and varying interest in running nodes over time, the proof of work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases.

5. Proof Of Tally

To implement a voter identity system that keeps track of address sending votes for election fraud management, we use a Proof of tally for every Bitcoin address used for voting. Using AXIOMITY we input a tally count into the blockchain for each voting address every time it votes, giving a timestamped tally to each vote done by that voting address. The Proof of tally is read by both AXIOMITY for voter verification and by the election for election data recording on each voter. The election can keep track of the Proof of tally on each address using it for results within the election. Every VOTE is given to a single Bitcoin address as a Edge Security login system. That address now gets hashed in a transaction with its Proof of tally number. The Smart Contract election looks for this Proof of tally and records it in the election. Each election is now connected to a voter's address and a tally count that goes up on each vote completed. This gives a profile of a Bitcoin address as Active Status, Inactive Status or Fraudulent Status, allowing AXIOMITY to accept or reject the voter as a real voter or a fraud. The elections record the Proof of tally on each address showing a history of votes from a voter while keeping the actual person behind the Bitcoin address private.

6. Candidates, Legislation & Amendments

Using the Legislation & Voting tool called AXIOMITY, one can hold their votes securely using a private key like a Bitcoin wallet. Within AXIOMITY there are several features, from creating elections to creating legislation. Anyone can create a piece of legislation and post it to the election, viewable and interacted with by AXIOMITY, other AXIOMITY users can select that piece of legislation now viewable to the public, select any word, sentence or entire sections of the law and submit an amendment. This creates a new piece of legislation and posts it to a new election. All amended legislation is branched in a history viewable in AXIOMITY. Every piece of legislation can be viewed, amended and voted on by anyone holding a vote unless it's created as a private election, this allows only specified vote addresses to vote. When a piece of Legislation is posted a new election is created on a Smart Contract with its custom set of rules implied by AXIOMITY. When a candidate or piece of legislation is voted for yay or nay, a VOTE, a Counterparty token, is sent from the voter to the election, the VOTE is immediately returned to the Voter when the election ends. bitcoin data is sent by Axiomity to activate the Smart Contract election holding the piece of legislation. Posting a Vote casts a yay or nay vote & the legislation gets a vote count up or down +1 or 0. Depending on the vote count in the election when it ends, the registered winner is sent a winning token using Counterparty and the legislation moves up in ranking on a public common law ledger similar to a blockchain explorer depending on its vote count, how many winning tokens it and its amendments have in total & a reddit like up and down popularity poll. The posted Candidates, Legislation & Amendments can all be seen in AXIOMITY the wallet/explorer, as it relays callbacks from Counterparty running over the Bitcoin Blockchain. It will allow anyone to post as a candidate or law in a custom election, with all changes to legislation, all votes and all events timestamped into a blockchain of elections, holding votes and voters public keys within the elections history. All elections are hosted on the Smart Contract blockchain and will be interactive using callbacks, websockets, get, post & http requests.

7. Election

An election is created within a Smart Contract using AXIOMITY or some other blockchain system & user interface. Each election will have an election timed lifespan, set of rules, candidates, legislation, budget & an accessible URL that can be accessed by the public. Each election has its own Smart Contract address and using AXIOMITY communicates with the Counterparty address & Bitcoin Address that togbitcoin data holds and moves the votes from voter to candidate and then back to voter. Within an election each law can be voted for yay or nay. When a vote is casted a vote is sent from voter's address to legislations yay or nay addresses or to a specific candidates address, these addresses are built into the election smart contract. When an election has candidates or legislation receiving votes the election smart contract responds to and records every vote into the Bitcoin Blockchain using the Counterparty VOTE token. The election logs the changes, the vote count is recorded and displayed within AXIOMITY using bitcoin data onto the Smart Contract Blockchain. This allows a multiple blockchain record of the Legislation, Vote & Voter interaction. The election once expired will automatically return all votes casted, but will not increase the vote count of the legislation or candidates within the election.

8. Voting

When voting for a piece of legislation or candidate each yay, nay or candidate has an address. A vote token is sent from the voters AXIOMITY application to the address of the legislations yay nay or if a candidate their public address, the election maintains the votes until the election is ended, then the votes are returned to the voters. This way votes are recycled and personally held. A proof of tally system will allow each voter to build a tally count on the amount of elections they have participated in. This shows an address of a voter as active, non active or fraudulent depending on the proof of tally, a voter will be denied an election if the voter's address is found to be fraudulent based on double vote attempts from that address. VOTEs can be acquired by creating a profile on the BitCongress website. The only way to get a vote is to submit a Bitcoin address as a owner of the vote. Once a bitcoin address is given a VOTE, it will never be able to receive a VOTE again from the BitCongress site. This will allow for a truer Proof of tally to be used with confidence. VOTEs can be sent from Voter to Voter, they are automatically returned to the sender or voter, but there is a record of this on the Bitcoin Blockchain. This allows for the vote to not be used as a monetary tool, but a vote token for record of approval. It keeps a record of this to build a Proof of tally on all Bitcoin addresses used in elections. Showing a voter profile for Bitcoin addresses, yet holding that voters privacy.

9. Count Methods

One of the most important features of BitCongress is the count method used for massive elections such as presidential, legislation that millions of people will be voting on does not constitute a normal vote count. We propose the use of the Borda Count in a modified manor married to a blockchain of consensus. The Borda Count allows for a points system of election over a normal standard election count. In the US Presidential elections there is a system used called the electoral college. This system allows an institution of electors to pick a candidate depending on the 270+ count. If the electoral college cannot elect a winner, the House, then the Senate votes for president. We propose a count system that uses a Modified Borda Count as a primary count for all elections by default & a Quota Borda System for any “large” scale elections. If this count method is not fulfilled a winner a secondary vote can be done using a different count method. The Borda count is a single winner election method in which voters rank options or candidates in order of preference. The Borda count determines the outcome of a debate or the winner of an election by giving each candidate, for each ballot, a number of points corresponding to the number of candidates ranked lower. Once all votes have been counted the option or candidate with the most points is the winner. Because it sometimes elects broadly acceptable options or candidates, rather than those preferred by a majority, the Borda count is often described as a consensus based voting system rather than a majoritarian one. The Modified Borda Count is used for decisionmaking. For elections, especially when proportional representation is important, the Quota Borda System is used. Under the Borda count the voter ranks the list of candidates in order of preference. So, for example, the voter gives a '1' to their first preference, a '2' to their second preference, and so on. In this respect, a Borda count election is the same as elections under other ranked voting systems, such as instant runoff voting, the single transferable vote or Condorcet methods.

10. Network

The steps to run the network are as follows: Within Bitcoin & Counterparty new transactions are broadcast to all nodes. Each node collects new data from bitcoin and counterparty transactions such as VOTES, BTC, XCP or other tokens and the token note data into a block. Each node works on finding a difficult proof of work for its block, for the election as running & a Proof of tally of voters within the election to help authenticate the voter. When a node finds a proof of work, it broadcasts the block to all nodes. Then AXIOMITY finds the Proof of tally for the address that sent the VOTE and accepts active status, inactive status and rejects fraudulent status addresses. Nodes accept the block only if all transactions in it are valid. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash. Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proofofwork is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one. New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one. Counterparty running over the Bitcoin Blockchain gives each piece of legislation a yay & nay address and each candidate an address. The VOTE is a token created by Counterparty and thus uses the Bitcoin mining system, but has no monetary value as it is limitless and returns to sender after an election. The election is a smart contract created on the Smart Contract Network. bitcoin data and counterparty data/tokens is sent to the election from AXIOMITY to create and start a new election. Bitcoin data and counterparty data/tokens then sent to a winner contract that executes a rule to return the VOTEs back to all voters, while the winner contract registered the winner and sends the bitcoin data back to the AXIOMITY client. This allows for bitcoin data, VOTE & Bitcoin to be recycled through elections, voters and voters, all the while posting all results on the corresponding Blockchains. AXIOMITY as the front end wallet, law creation & voting application holds XCP, bitcoin data, BTC, VOTE & uses them all in sync and combination with their respective blockchains.

11. Incentive

Using Bitcoin, Counterparty & Smart Contracts as Blockchains to record, hold and move data, we can depend on the mining ecosystem of these Blockchains to give the system functionality without creating a cryptocurrency with monetary value. VOTES are limitless and are returned to voter after election, giving incentive to vote, not to buy & sell VOTES. For monetary incentive the Bitcoin system is used as an underlying budget & infrastructure for BitCongress along side the Counterparty network for token creation and distribution within the Bitcoin Blockchain. In Bitcoin, by convention, the first transaction in a block is a special transaction that starts a new token owned by the creator of the block. This adds an incentive for nodes to support the network, and provides a way to initially distribute votes into circulation, since there is no central authority to issue them. The steady addition of a constant of amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is CPU time and electricity that is expended. The incentive can also be funded with transaction fees. If the output vote count of an election is less than its input vote count, the difference is a election fee that is added to the incentive value of the block containing the election. Once a predetermined number of votes have entered circulation, the incentive can transition entirely to election fees and be completely inflation free. The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU

power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the validity of his own use. Votes are automatically returned to voters after an election, so every vote, has monetary value, identity information & multiple uses in multiple elections. The election fee is equivalent to the monetary cost of running a pencil, paper, brick & mortar & human vote counter based election. Elections are timed and when they end all votes are returned to voter's public key. After every election a public record of all voting activity is held on a blockchain.

12. Simplified Token Verification

It is possible to verify votes without running a full network node. A user only needs to keep a copy of the block headers of the longest proof of work chain & the longest Proof of tally chain from that voter, which he can get by querying network nodes until he's convinced he has the longest chains, and obtain the Merkle branch linking the VOTE to the block it's timestamped in. He can't check the election for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it. As such, the verification is reliable as long as honest nodes control the network, but is more vulnerable if the network is overpowered by an attacker. While network nodes can verify elections, VOTES and voters for themselves, the simplified method can be fooled by an attacker's fabricated elections, VOTES and voters for as long as the attacker can continue to overpower the network. One strategy to protect against this would be to accept alerts from network nodes when they detect an invalid block, prompting the user's software to download the full block and alerted transactions to confirm the inconsistency. Individuals, Schools, Businesses or States that hold frequent VOTEs will probably still want to run their own nodes for more independent security and quicker verification. Within the election the smart contract can interact with callbacks, tokens and its native bitcoin data. Within AXIOMITY a confirmation count is shown on both Bitcoin for fees, Counterparty for VOTES & Smart Contract for bitcoin data, these tokens & cryptocurrencies are all verified by their specified miners & auditing pools and cross referenced from the election to the voter within AXIOMITY. 9. Combining and Splitting Value Although it would be possible to handle VOTES individually, it would be unwieldy to make a separate transaction for every portion of a vote in an election. To allow votes to be split and combined, transactions contain multiple inputs and outputs. VOTES are divisible like Bitcoins (1.00000000), this allows for a more custom implementation of a vote count within a private or small election. Normally there will be either a single input from a larger previous elections or multiple inputs combining smaller amounts, and at most two outputs: one for the candidate or legislation, and one returning the vote to the voter after the election. It should be noted that fanout, where an election depends on several elections, and those elections depend on many more, is not a problem here. There is never the need to extract a complete standalone copy of a elections history.

13. Privacy

The traditional centralized voting model even with electronic voting systems achieves a level of privacy by limiting access to information to the parties involved and the trusted third party. The necessity to announce all elections publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. The public can see that someone is sending a vote to a candidate or piece of legislation, but without information linking the vote or election to anyone voting. This is similar to the level of information released by stock exchanges, where the time and size of individual trades, the "tape", is made public, but without telling who the parties were. As an additional firewall, a new key pair should be used for each election to keep them from being linked to a common owner. Some linking is still unavoidable with multi input transactions, which necessarily reveal that their inputs were owned by the same owner. The risk is that if the owner of a key is revealed, linking could reveal other elections, legislation, amendments and votes that belonged to the same owner. For this we propose the second output be the voter itself as the election has a timed contract and when it ends, the election holding the votes returns them to the voters. With this a voter can reuse the public key over and over again, building its Proof of tally. This helps the public give a numerical count to the public keys vote use but not its vote history.

14. Conclusion

We have proposed a system for electronic election, legislation & voting without relying on trust. We started with the usual framework of Bitcoin, Smart Contracts, Counterparty & Blockchain technologies. The system allows for a easy creation, post & ranking of legislation through a mobile application, timestamped and hashed by Bitcoin, managed by Counterparty & held in Election by Smart Contracts. This allows a combination of technologies to spread the decentralized notion of vote and election as far from centralized systems as possible. Incentive is from Bitcoin mining fees, Smart Contract fees, Counterparty reward tokens for building infrastructure for BitCongress & its case use for the world's first Blockchain Voting System that will be able to compete with the current legislative and governing systems. We create a token on Counterparty called a VOTE and it is sent to legislation or candidates, recorded on the blockchain and then returned to the voter, giving a registered endorsement on a blockchain. We create a Legislation, Amendment & Voting wallet called AXIOMITY to hold bitcoin data, Bitcoin, Counterparty & our VOTE token, it allows creation of legislation, amendment of posted legislation, debate, qurum, filibuster and voting in all public elections. It allows the creation of private elections corresponding to designated voter addresses. BitCongress is a platform and process based on technologies, concepts and systems running on Blockchain Technologies, Timestamps, Open Source code & decentralized based principles.