Election Integrity through the Blockchain: Sprint 1 Summary

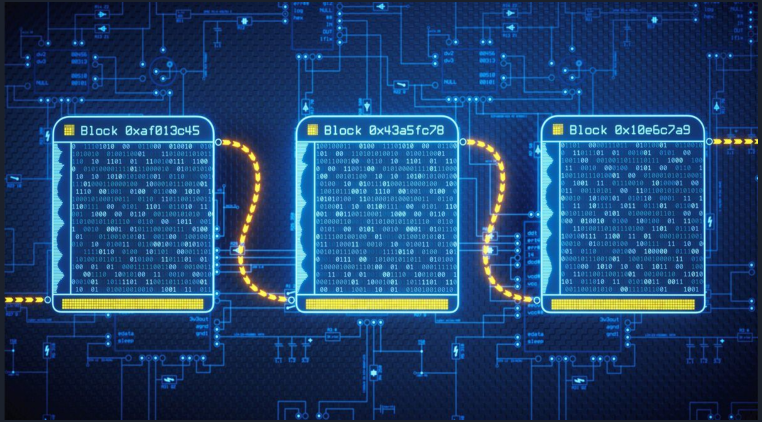
Samir Farhat Dominguez
U41707119
10/12/2021

Product Mission

- Initial motivation: US 2020 elections and January 6th, but applicable universally
  - Dominion hardware vulnerabilities
  - Data mining algorithms and statistical methods showing infeasible probabilities
  - Corruption and election interference is inevitable, but it can be made small enough to prevent outcome changes
    - Millions of dollars to change outcome of an event that influences the movement of billions or even trillions of dollars
- How does the state, regime, or sovereign govern
  - Monopoly on force: Police and Military
  - Monopoly on legitimacy: Societal acceptance(social contract) and civic convention
- Blockchain verification focuses on the latter
  - Creates a transparent, accessible, verifiable, immutable framework by which citizens are able to cast a single vote each, and have that vote private while still being attributable to a legitimate voter
  - Decentralized, and therefore not subject to corruption or manipulation
  - Cheaper to maintain and run then current systems

Literature Review

- Blockchains are cryptographically secure, distributed ledgers which can hold records on any smart contract that is called on the blockchain
- It is a sequence of blocks linked cryptographically, usually by hashing the timestamp associated with each new data creating event.[1]
- These blocks each contain information on any supported smart contracts called on the chain, most commonly individual transactions and movements of the cryptocurrency they maintain
- Blockchains differ in complexity and support different mechanisms
  - BTC supports token transactions only pretty much
  - ETH supports NFTs, storage, dapps, virtualisation, tokens, governance and many others



MVP

- The idea for the MVP would be to have a system where users are able to manually submit credentials, including proof of residence and ID alongside their wallet’s public key.
- A human(maybe AI in the future) would corroborate these submitted credentials, and send a smart contract transaction whereby a VOTE token is issued alongside a predetermined quantity of gas fees to the user wallet.
- The user would then be able to interact with a system by which they sign this token as an NFT with their vote and submit that through a smart contract to a dispatch system
    - For purposes of the MVP the mass submission of credentials will be simulated as a directory full of JSON files with voter credentials
    - The actual dispatch of the VOTE token will occur on the blockchain testnet, so tokens and gas will be issued to dummy wallets I have access to on the testnet. This will require a smart contract to be developed on the chosen blockchain(NEO for now)
    - The actual sending of the vote by the user will also occur on the blockchain and a smart contract will have to be able to do this and…
        - Restrict the receiver of the token to my testnet dispatcher wallet,
        - Dispatcher will only accept 1 token from a wallet address

User Stories

- I can send my credentials as PDFs and text file(for public key and wallet address) and receive a confirmation on my email(only implementing gmail support on my MVP)
  - Script checks files not empty or null and if public key is valid and wallet address is valid
- I can receive my token and gas fees to my specified wallet
- I can sign and mint my NFT and deploy the cast vote smart contract

Technologies

- Blockchain must support
  - In chain, third party tokens
  - Token distribution
  - Token ownership
  - Public block explorer and private address system.
  - Non-fungible token attribution.
- Bitcoin does not
- Ethereum
  - ERC-20 protocol is the most developed, robust, matured, and proven blockchain system out there.
  - Allows for on-chain third party tokens, token distribution, token ownership, public block explorer, non-fungible token attribution and private address system. A
  - ETH, is notoriously expensive and block times are rather slow(Not relevant for MVP).
- NEO
  - NEO is not as developed as ethereum, but still contains the necessary services to develop the voting system
  - Its gas fees and block times are exceptional
  - Block explorer API would be tremendously useful to create a voting transparency application.

Dev Environment

- Github for version control and publishing
- Visual studio code for C# development of smart contracts(native language of NEO blockchain)
- Pycharm for Python verification script
- Ubuntu 20.04 OS to run test network
- NEO blockchain test network and utilities