# 

# Final Project

Guy Bar-Yosef and Shifra Abittan

## 

Table Of Contents

[**Final Project**](#_73ea642pr7sy) **1**

[Project Details](#_4gfsnep4pqe) 3

[Abstract](#_f74j6xi9f2d) 4

[Background](#_6uzztn1xet7w) 5

[Methods](#_7o1gerjbd2wm) 7

[Schedule](#_obw9ta5gbak5) 9

[Discussion](#_jc0yxfaqn6hr) 10

[Deliverables, Results, and Progress Maintenance](#_5jmglqqg3jjd) 12

[References](#_55nuciea96g3) 13

## 

## Project Details

This project hopes to incorporate aspects of multi-agent systems into a free and open-sourced dApp built on the Ethereum Blockchain network. It will utilize and draw inspirations from several other projects including uPort, Truffle, and The Open Vote Network, and will serve as both a useful resource for the community as well as a learning experience for us.

A rough sketch for each member’s contributions:

**Shifra:** Integrate uPort, Back-End, Integrate off-chain database, Smart Contracts

**Guy:** Integrate zkSnarks, Front-End, Integrate off-chain database, Smart Contracts

## 

## Abstract

Voting has historically been a centralized process, with a trusted intermediary collecting participants’ votes and publishing the result. Inherently, participants must trust the integrity and honesty of the intermediary party, which introduces a single point of failure into the system should the third party choose to default. As Joseph Stalin famously remarked: “The people who cast the votes decide nothing. The people who count the votes decide everything." This highlights how throughout history, this point of failure has been exploited continuously. The masses recognize this failure and often accuse illegitimacy, as seen as recently as in the last United States presidential election. By using a blockchain, the voting process can become decentralized, replacing the single point of failure with a trustless entity that exists on a secured, distributed system. This secure system will allow for greater flexibility, transparency and trust in the outcome. As a society, voting on the blockchain can offer all participants the ability to propose personalized voting instances and revolutionize the way democracy is exercised.

## 

## Background

The procedure of voting is one in which a multitude of individuals choose from an array of choices, and a voting algorithm is applied to select one such outcome. Anonymity is often desired; each participant’s choice should be kept secret from the other parties. In addition, each participant’s vote should be weighted equally. Such a voting procedure is used in a plethora of scenarios, ranging from political elections and appointments to small scale decision making within a company or institution.

The voting process, for centuries, has required the use of a centralized body to serve as the clearinghouse to tally the ballots and announce the results. This implicitly demands that all participants place trust in the centralized body. All voters must trust that the centralized body is not acting honestly: not tampering votes, eliminating votes and fabricating votes to align the outcome with a particular bias. Finding a trustworthy third party is a tremendous hurdle and places the entire voting procedure in a state of suspicion. By utilizing a decentralized blockchain to implement a voting framework, the third party is eliminated, resulting in a trustless procedure.

Because of the strong need for a blockchain-based voting system, multiple applications have already been built. The most prominent application is called Follow My Vote[[1]](#footnote-0). Although the application promises to deliver a “secure and transparent voting solution,” to deploy the software is not simple. One must contact the company first to acquire access to the software, which involves a layer of centralization. In addition, all of the links and buttons on the website are broken.

Since 2005, the country of Estonia has conducted their nationwide political elections over the internet using a blockchain, referred to as “i-Voting.”[[2]](#footnote-1) Cybernetica[[3]](#footnote-2) is the company that built the technology framework. Although the voting is stored using a blockchain, the process of ensuring “one person, one vote” and verifying identity is done exclusively with hardware. Our product will deploy cryptographic software to achieve the same results, which is cheaper. It also allows any private individual to run an election easily, without having to distribute highly specialized hardware to all of the voters.

We aim to build a secure, cheap voting decentralized application that can be utilized by any individual for any election, small or large.

## 

## 

## Methods

Allowing society to exercise its democracy, voting is among the most fundamental and important social events. The importance attributed to this event has also made it a major aspect of multi agent research, as the large amount of complex interactions can lead to strategic outcomes.

This voting platform includes several implementational challenges, including how to enforce correct participation and eligibility of users, how to process a large volume of votes, and how to confirm that once a participant has voted that their actual vote remains anonymous. As a response to the identity problems that arise when digitizing the voting process, the voting dApp will use UPort to govern and confirm the participant’s identities. UPort’s SDK will serve as a handle to confirm who and how many times different participants are voting.

The issue regarding how to confirm a participant has voted without intruding on their privacy and revealing their vote will be solved using zero-knowledge proofs. This cryptographic method will be written onto an off-chain database and will serve as the actual casting of the vote.

With the voting methodology cover, the next concern that will be addressed is the scalability issues that arise when there exists a large amount of voting participants. Besides burdening the Ethereum network, these transactions will quickly become ficially unrealistic and so as hinted at before, the proposed solution is to move the voting process as off-chain as possible while still retaining the immutability advantage that Blockchain technology provides. Therefore the voting will happen as a zero knowledge proof that is being written into an off-chain database. Once the time allotted for voting has run out the dApp will hash all of its relevant zero-knowledge proofs and compute the results.

## 

## Schedule

## 

## Discussion

*Evaluation*

In order to evaluate the success of the project, a voting instantiation will be demoed. A successful scenario of the application will include the following chain of events: (1) One participant will deploy an “election,” choosing from a host of features, including duration of time before the vote closes, which will be implemented using locktimes in the block header. (2) Individual participants will be granted the right to vote, either by sending a link to them or by directly adding the uPort identities. (3) Each participant that was invited, and desires, to vote will cast their vote. (4) Some voters will change their vote multiple times before the end time of the election, in order to showcase that only the last vote cast by each participant is counted. (5) Software will parse all of the votes, employ a particular multi-agent voting technique to choose a result. (6) A hash of the votes will be uploaded to the blockchain. The reason that the blockchain is only used at this step is to ensure that the cost of using our dApp is minimal. As the size of the data uploaded to the chain grows, the price quickly escalates. Therefore, only a hash of the aggregation of voting results will be uploaded. This will allow for cheap use of our platform.

A transaction will include one hash of all of the votes cast by the participants and the outcome of the election. The transaction can easily be verified by hashing the voting data and ensuring that it matches the hash stored in the blockchain. This verification process must allow the verifier to access the voting data without being able to “see inside of it.” The verifier must be able to see that a participant voted, without gaining access to the contents of the vote. ZKSnarks, a method of zero-knowledge proof, is how we hope to accomplish this. The advantage of using a blockchain to store the hash is the property of immutability. Once the election is closed and the hash of the votes and outcome is uploaded to the blockchain, no participant can tamper with the votes or results without changing the hash value.

We plan to consult with two experts from Ethereum: Jessica Marshalls and Ben Siegel.

*Challenges*

One low hanging fruit is the multi-agent system voting protocol implementation. Using a series of if statements, it should not be difficult to write code to loop through all of the results of the election and process and choose an outcome.

After speaking to Jessica Marshall, one particular learning curve that she alerted us to is writing a smart contract. She warned that it is often quite difficult to write a smart contract for the first time, so this may present itself as an area of challenge, although a surmountable one.

## 

## Deliverables, Results, and Progress Maintenance

As per the schedule above, deliverables for this project will include weekly updates of progress as well as a final presentation on the 9th of May. This final presentation will include a complete summary of the project including its successes, and failures, and will also contain recommendations for future steps and areas of explorations. As part of this final presentation our group will present how a voting instance is instantiated, perhaps even using the class as the participants, and show how the vote is conducted and added to the blockchain. We will highlight the advantage that the Blockchain provides in terms of its immutability and show the platforms main features such as the privacy that that is provided by zero knowledge proofs.

We will also have a more thorough showcasing of our project’s features, such as its uPort integration, in youtube videos that will be available via the project’s website, a page that will doubly serve as both its main page, and the application’s actual user interface, thereby making it easy and intuitive to use.

Throughout the project’s lifecycle it will be maintained on GitHub, with an README file that will be constantly updated to reflect progress, bugs, and future steps. This project is meant to be free and open-sourced, hopefully a useful addition to a growing and supportive community, and as such we hope to make it as widely available as we can.

## 

## References

Anon. Open Identity System for the Decentralized Web. Retrieved April 8, 2018 from https://www.uport.me/

Anon. Your Ethereum Swiss Army Knife. Retrieved April 8, 2018 from http://truffleframework.com/

Mike Summers. 2016. Online Voting Isn't as Flawed as You Think-Just Ask Estonia. (October 2016). Retrieved April 8, 2018 from https://spectrum.ieee.org/telecom/internet/online-voting-isnt-as-flawed-as-you-thinkjust-ask-estonia

Patrick McCorry, Siamak F. Shahandashti and Feng Hao. "A Smart Contract for Boardroom Voting with Maximum Voter Privacy". *Financial Cryptography, 2017*.

Timothy Ko. 2018. A guide to developing an Ethereum decentralized voting application. (March 2018). Retrieved April 8, 2018 from https://medium.freecodecamp.org/developing-an-ethereum-decentralized-voting-application-a99de24992d9

1. https://followmyvote.com/ [↑](#footnote-ref-0)
2. https://e-estonia.com/solutions/e-governance/i-voting/ [↑](#footnote-ref-1)
3. https://cyber.ee/en/e-government/i-voting/ [↑](#footnote-ref-2)