Kantonsschule Enge
Zürich

Building a Decentralized E-Voting Application on Ethereum

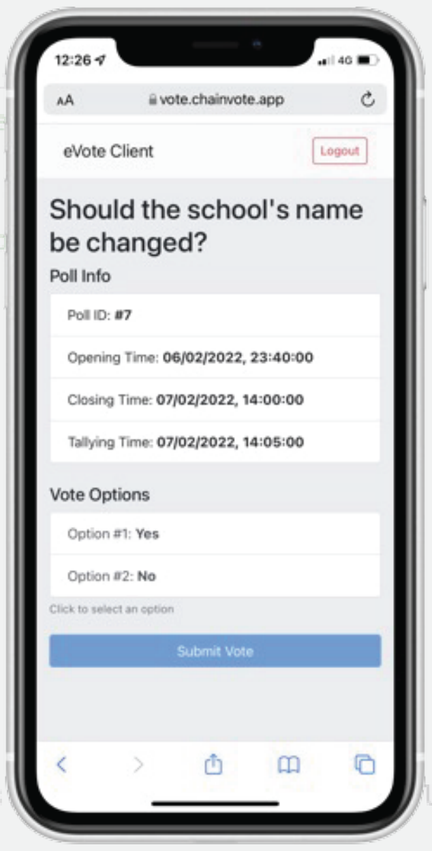
Introduction

Since the launch of Ethereum in 2015, which made it easily possible to also run applications on blockchains, so-called decentralized applications have become more and more popular. A few years later, in 2019, suddenly both available e-voting systems in Switzerland were shut down, meaning that there currently aren't any e-voting solutions available in Switzerland. With the rise of decentralized applications and the need for secure e-voting systems in Switzerland, the following question arises, which the paper tries to answer:

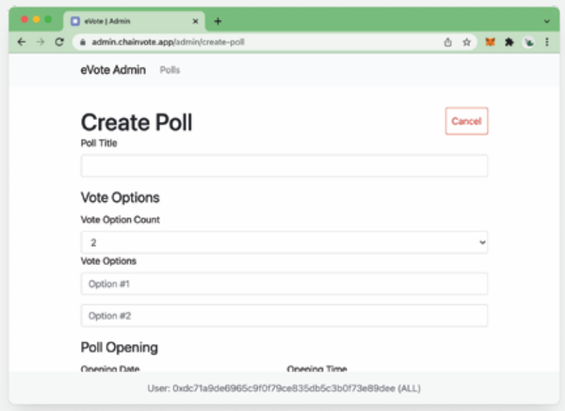
Is it possible to build a decentralized e-voting system that can be used for governmental elections in Switzerland on initiatives and referendums?

Process

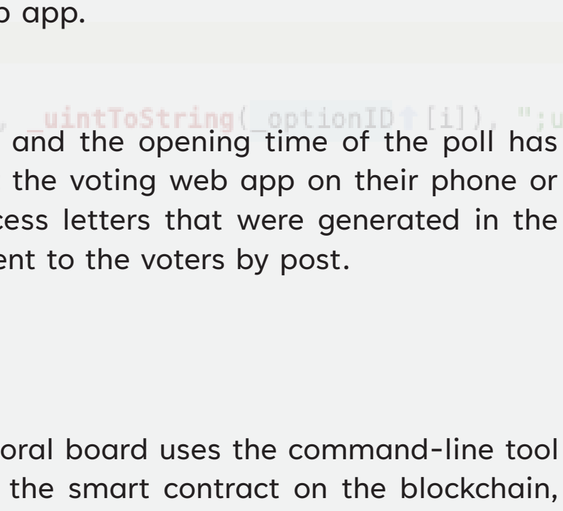
The practical process in the paper can be divided into four parts, between which there were also overlaps. The first part consisted of learning the Solidity programming language, which is necessary to program the smart contract, the central component running on the blockchain. In the second part, the legal and practical requirements that an e-voting system has to fulfil were elaborated. Based on these requirements, I then designed my own e-voting system, by working out all the necessary processes, security measures, applications and more. In the fourth part, all applications necessary for the e-voting system were programmed.



Voting Web App



Admin Web App



Registration App

Administration

The e-voting system is managed by 12 administrators with different roles and capabilities. There are 3 registrars, 3 chairpeople and 6 electoral board members. For some important actions, there is even a two-thirds majority inside the different admin groups required. These measures are primarily to prevent the abuse of power by a single administrator or even a group of administrators and to enforce checks between the different administrators.

Election Stages

When conducting an election with the e-voting system, there are six different stages.

1. Encryption Keys are Generated

In the first stage of the election, the electoral board uses a command-line tool to create the encryption keys, which are later used in the voting phase to encrypt the votes. This is necessary because all the data on the blockchain is publicly accessible, but the votes need to stay private until the election ends.

2. A new Poll is Created

In the second stage, the chairpeople fill out a form in the admin web app, where they need to specify the title of the new poll, the available vote options and more, including the public key generated in the first step. When the form is submitted, a transaction is sent to the smart contract on the blockchain, which then creates a new poll.

3. Voters are Registered

In the third stage, the registrars first use a desktop registration app to create the voter accounts and the access letters, which contain the "passwords" of the voters. In a second step, the voters are registered, verified and confirmed by the chairpeople with the admin web app, which communicates with the smart contract.

4. Poll is Verified

In the final stage, before the voters can vote on the poll, the electoral board verifies that the details of the poll are correct. They check that the vote options, opening, closing and tallying time and the other details of the poll, including the voters, are correct. If this is the case, they confirm the poll in the admin web app.

5. Voting

Once the poll has been approved and the opening time of the poll has been reached, the voters can visit the voting web app on their phone or computer and log in with the access letters that were generated in the third stage and have since been sent to the voters by post.

6. Votes are Counted

After the poll has ended, the electoral board uses the command-line tool to decrypt and send the votes to the smart contract on the blockchain, where they are publicly counted. Only all the electoral board members together can do this, meaning that no member alone can decrypt the votes before the election ends.

Source Code

To view the source code of the e-voting system, scan the QR code or visit:
bit.ly/evotesystem-code

Matura Paper by Levin Heimgartner
Supervised by Patrik Marxer