Electio - User Manual

Patrick H Morris

Advisor: Dr. Geoff Hamilton

Student ID: 14759021

May 19th, 2018

Contents

1 Introduction

The purpose of this document is to outline the steps involved in installing and using the Electio application. Electio is intended to be used in a Linux environment as much of the software underpinning Ethereum is developed with Linux in mind.
The steps which are instructed in this document are a reflection of the project as a proof of concept and as of current standing does not represent a fully-fledged application. The intended usage of this project in its current state is in a local environment with which I have enabled election processes to be simulated.
A version of this project can be found at

http://46.101.45.240:3000

In order to use the system at this address, the mnemonic key below is necessary. How to use it will be shown further in the document

```
Mnemonic:

foster decide decline bridge
behave clarify click final
dolphin universe february auto
```

2 Installation

Below I will go through the steps required in order to setup and install the electio application. First clone or download the project via gitlab using this command:

```
$ git clone https://gitlab.computing.dcu.ie/morrip25/2018—ca400—morrip25.git
```

2.1 Hardware Requirements

This is a list of the mininum hardware requirements that I would advise anyone who wishes to use this application to use.

- OS : Ubuntu 14.04/16.04 LTS
- RAM : 4gb+
- Disk space : 10gb+
- CPU : Intel i3 or AMD equivalent

2.2 Software Requirements

This is the list of the required software that a user must install to use this application in their local environment. As the levels of development in the Ethereum blockchain space is quite rapid, many of these packages are still in development and are not fully fledged tools. This led to some components of the application becoming deprecated such as with MetaMask.

- Nodejs v8.11.1
- npm v5.6.0
- Google-Chrome web browser
- truffle v4.0.0 (deprecated)

- ganache-cli v7.0.0-beta.0
- metamask* v4.4.0 (deprecated)

The MetaMask tool is a google chrome plugin which is used to make transactions to the blockchain. The working version for this application can be installed manually by these steps:

- Navigate to chrome://extensions/ via the options menu or the web address bar
- Select developer mode in the extensions menu
- Select Load Unpacked on the menu bar and navigate to the electio project folder
- Select the metamask-build folder and metamask will be installed

The final step in the installation is to navigate to the project folder and simply install all packages defined in the package.json file:

```
$ npm install
```

3 Configuration

3.1 Blockchain Environment

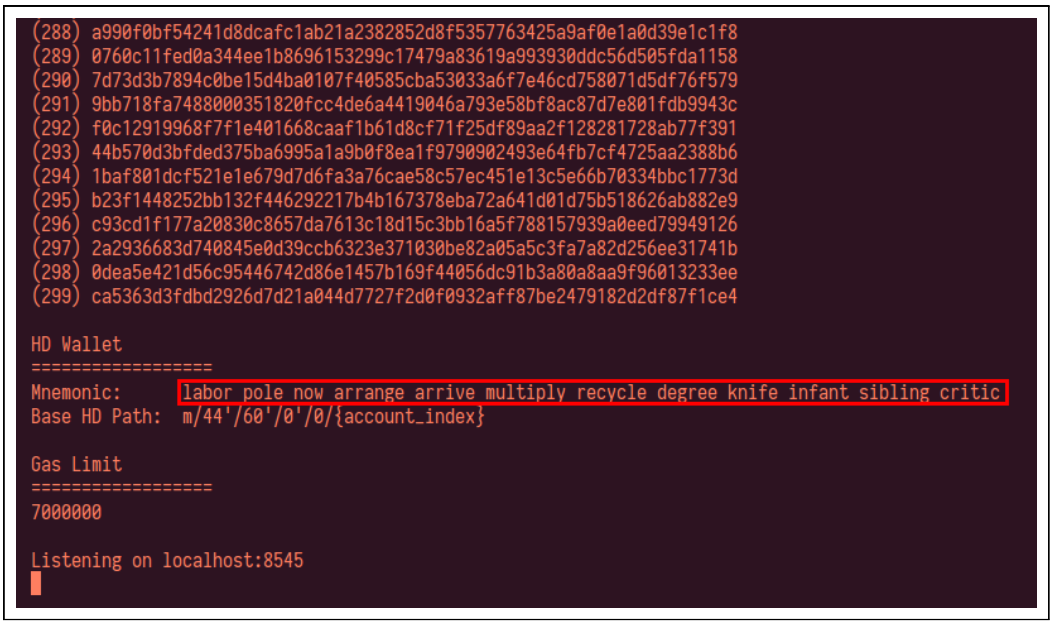
The backend of Electio is comprised of the Node.js server and the Ethereum blockchain node. The blockchain node for this application is a command-line tool called ganache-cli which simulates a blockchain environment locally. The ganache-cli can be started by simply calling

```
$ ganache—cli
```

But I find it preferable to call it with more arguments which I will explain

```
$ ganache—cli —a 300 —e 100000 ——gasLimit 7000000
```

This declaration will generate a blockchain of 300 accounts, each with a 10,000 worth of ether and a network which accepts cost of execution (gas) at a value of 7,000,000. This would then produce a result as below:

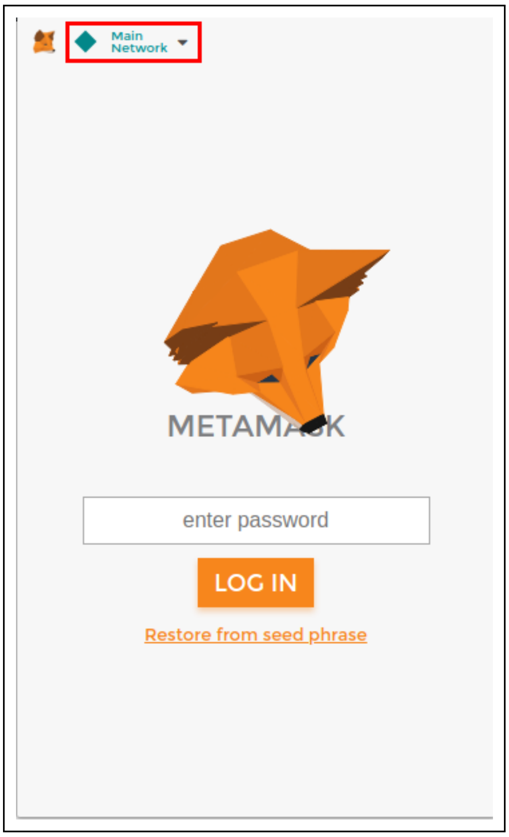


```
(288) a990f0bf54241d8dcafc1ab21a2382852d8f5357763425a9af0e1a0d39e1c1f8
(289) 0760c11fed0a344ee1b8696153299c17479a83619a993930ddc56d505fda1158
(290) 7d73d3b7894c0be15d4ba0107f40585cba53033a6f7e46cd758071d5df76f579
(291) 9bb718fa7488000351820fcc4de6a4419046a793e58bf8ac87d7e801fdb9943c
(292) f0c12919968f7f1e401668caaf1b61d8cf71f25df89aa2f128281728ab77f391
(293) 44b570d3bfded375ba6995a1a9b0f8ea1f9790902493e64fb7cf4725aa2388b6
(294) 1baf801dcf521e1e679d7d6fa3a76cae58c57ec451e13c5e66b70334bbc1773d
(295) b23f1448252bb132f446292217b4b167378eba72a641d01d75b518626ab882e9
(296) c93cd1f177a20830c8657da7613c18d15c3bb16a5f788157939a0eed79949126
(297) 2a2936683d740845e0d39ccb6323e371030be82a05a5c3fa7a82d256ee31741b
(298) 0dea5e421d56c95446742d86e1457b169f44056dc91b3a80a8aa9f96013233ee
(299) ca5363d3fdbd2926d7d21a044d7727f2d0f0932aff87be2479182d2df87f1ce4

HD Wallet
==================
Mnemonic:      labor pole now arrange arrive multiply recycle degree knife infant sibling critic
Base HD Path:  m/44'/60'/0'/0/{account_index}

Gas Limit
==================
7000000

Listening on localhost:8545
```

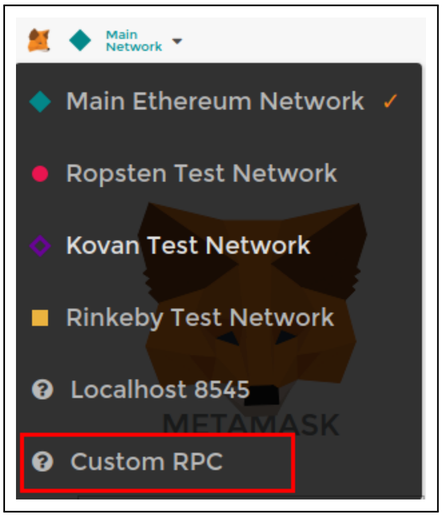
Highlighted in red is the account mnemonic which is used to generate the accounts. In order for us to use these accounts we copy this mnemonic into our metamask account plugin. We also must connect our metamask plugin to identify the local blockchain node

3.2 MetaMask Integration

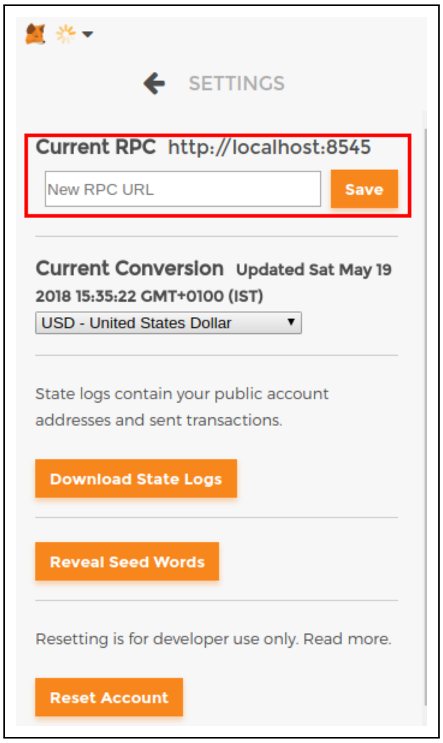
Metamask is a **wallet** which enables a user who has an ethereum account to interact with the blockchain by sending ether to other ethereum accounts. This system is crucial in dapps as it allows users a somewhat easy way of connecting to the blockchain. In Electio, it is used as a method of authenticating users as everyone has a unique ethereum account, Electio does not need to have an authentication system of its own.



Select Network Dropdown menu



Select the Custom RPC menu



Current RPC http://localhost:8545

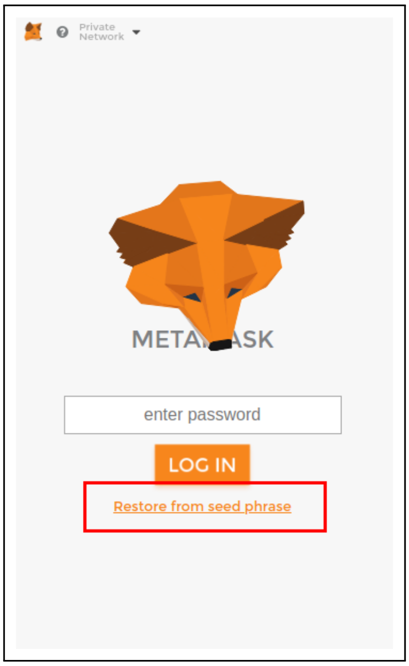
Current Conversion Updated Sat May 19 2018 15:35:22 GMT+0100 (IST)

USD - United States Dollar

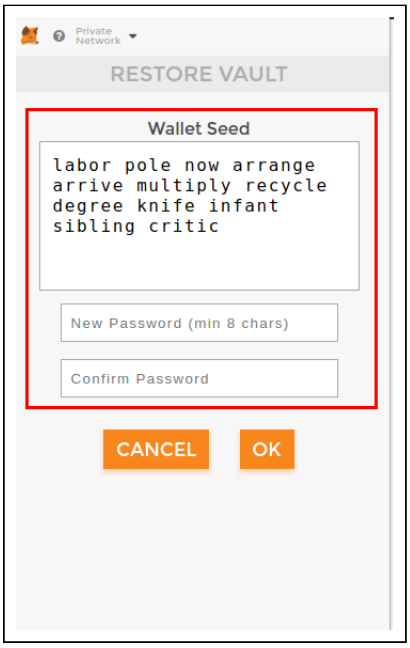
State logs contain your public account addresses and sent transactions.

Resetting is for developer use only. Read more.

Input the url http://localhost:8545 which will give a local network option for metamask to connect to.

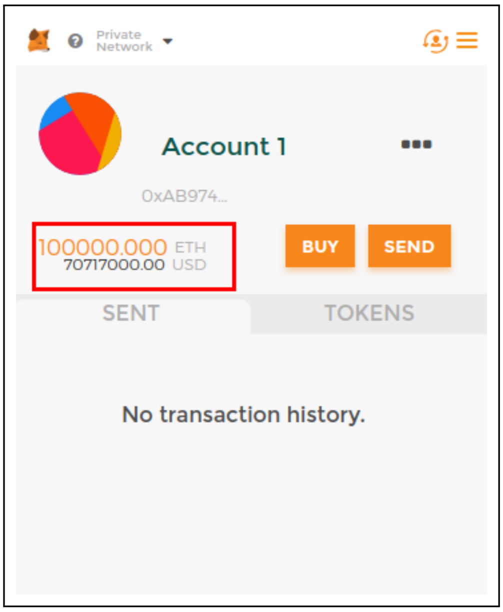


To generate the accounts from our local test blockchain we select to restore accounts from a seed phrase.



```
labor pole now arrange
arrive multiply recycle
degree knife infant
sibling critic
```

Copying it into our metamask account, a user must also select a user and password.



Account 1

0xAB974...

100000.000 ETH
70717000.00 USD

No transaction history.

The final result should look like this. To gain access to more accounts, select the option on the left of the hamburger and select create account.

3.3 Starting The Server

To start the server :

```
$ npm start
```

3.4 Contract Building & Deployement

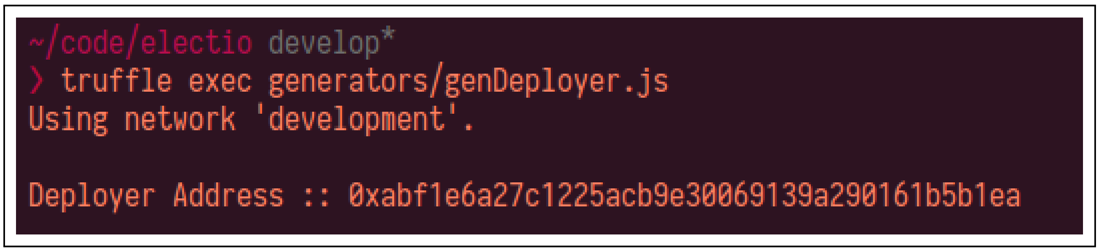
Smart Contracts are the logic that enables users of the application to use perform election functions on the blockchain. The Ethereum network can be defined as a global computer which allows people to instantiate a piece of logic which everyone can execute and achieve the same result from the execution. The gas which was alluded to in the ganache instructions is the cost of using this computer. The more work the logic does, the more gas it will cost but the benefit is that we can be assured as to what occurred as it is executed across the public domain.
We use truffle to create the environment to generate our contracts. These exist in the project folder under src/contracts. We build these using the command

```
$ truffle compile −−all
```

which compiles these down to binaries which then can be deployed on the blockchain. Truffle can be used to migrate these binaries but I have written my own scripts to streamline this better. They can be found under src/generators. There are two generator scripts which are used to create the election environment, genDeployer.js and genScenarios.js. Both deploy an instance of the main Electio smart-contract which is used to keep a record of all elections which are generated. These both return an ethereum address listing where the contract is located.
The difference between the two is that the scenarios script generates 9 scenarios to simulate the election process in different election systems and in different stages whereas the other will setup a blank Electio system.
To use, run :

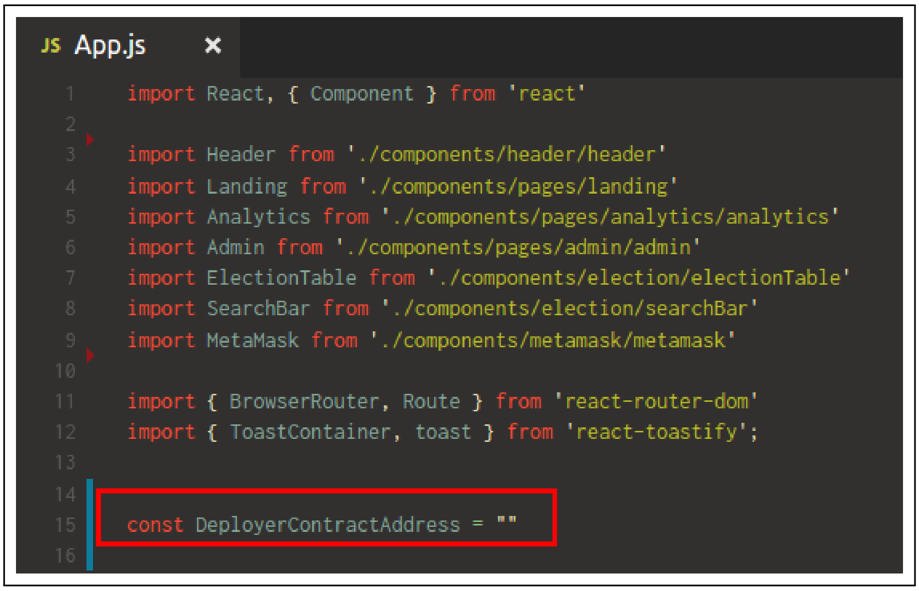
```
$ truffle exec src/generators/genDeployer.js
$ truffle exec src/generators/genScenarios.js
```

This will output a result like so:



```
~/code/electio develop*
❯ truffle exec generators/genDeployer.js
Using network 'development'.

Deployer Address :: 0xabf1e6a27c1225acb9e30069139a290161b5b1ea
```

This deployer address is to be copied into the file src/App.js : line:15



```
1   import React, { Component } from 'react'
2
3   import Header from './components/header/header'
4   import Landing from './components/pages/landing'
5   import Analytics from './components/pages/analytics/analytics'
6   import Admin from './components/pages/admin/admin'
7   import ElectionTable from './components/election/electionTable'
8   import SearchBar from './components/election/searchBar'
9   import MetaMask from './components/metamask/metamask'
10
11  import { BrowserRouter, Route } from 'react-router-dom'
12  import { ToastContainer, toast } from 'react-toastify';
13
14
15  const DeployerContractAddress = ""
16
```

At this point, the application is setup and ready to use. The frontend can be accessed via http://localhost:3000

4 Testing

Testing using this system is straightforward and can be done in a series of simple automated steps. I employed two methods in testing the application, unit testing and end-to-end testing which are written programatically using a modified version of mocha.js and chai.js which truffle provides.
Open two seperate terminals and in each respectively run from the project src directory:

```
$ ganache—cli —a 300 —e 100000 ——gasLimit 7000000
$ npm start
```

4.1 Unit Testing

The unit testing portion targets the basic functionality of the system. This involves the simulation of the contract functions which allow voters to register and deposit a vote. It also tests for the encryption, decryption and tallying utilities of the system. A large portion of the functionality is time-specific so the tests are carried out with a series of 1-5 second waits between some tests.
The units can be automatically tested by running:

```
$ truffle test src/tests/unit∗
```

4.2 End-to-End Testing

The End-to-End tests are generated in the same fashion as the unit tests but test a full simulation of the election process for the three electoral systems. They can be run using the command:

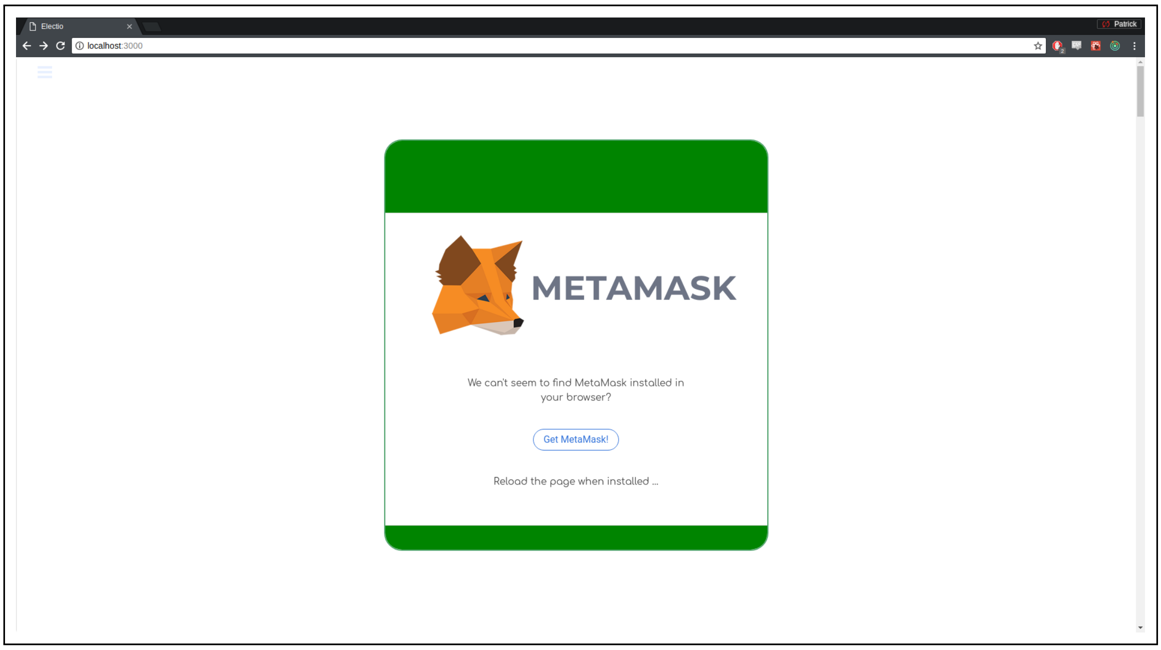
```
$ truffle test src/tests/e2e∗
```

Alternatively, all tests can be executed by running the command below from the src directory

```
$ truffle test
```

5 Usage

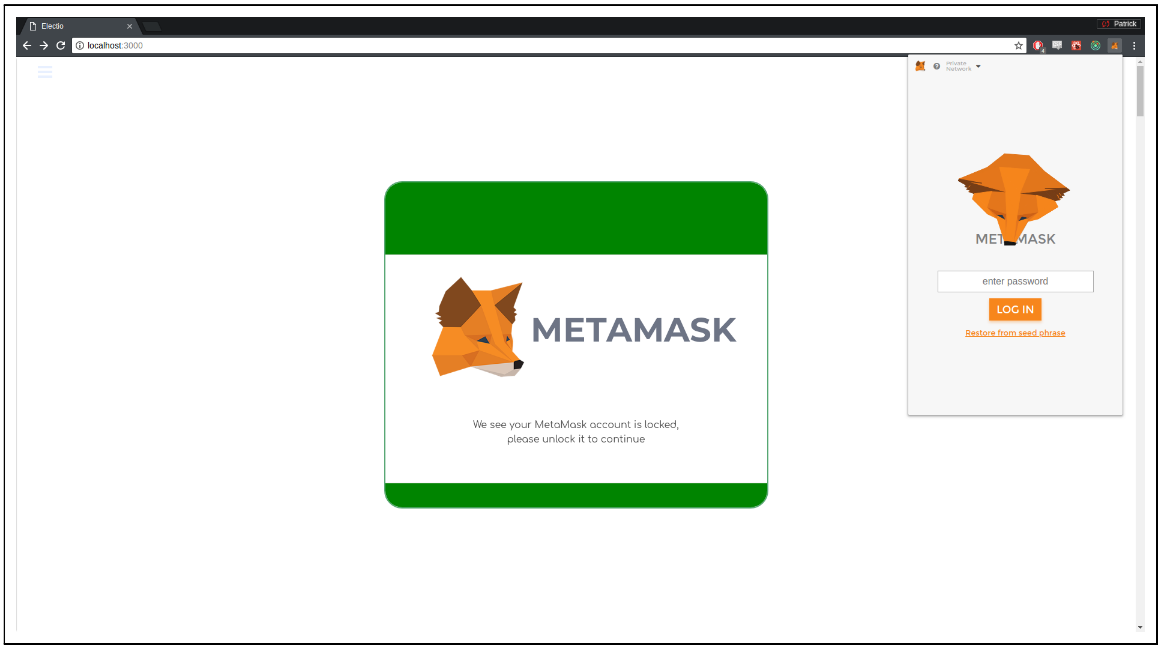
5.1 Authentication



We can't seem to find MetaMask installed in your browser?

Reload the page when installed ...

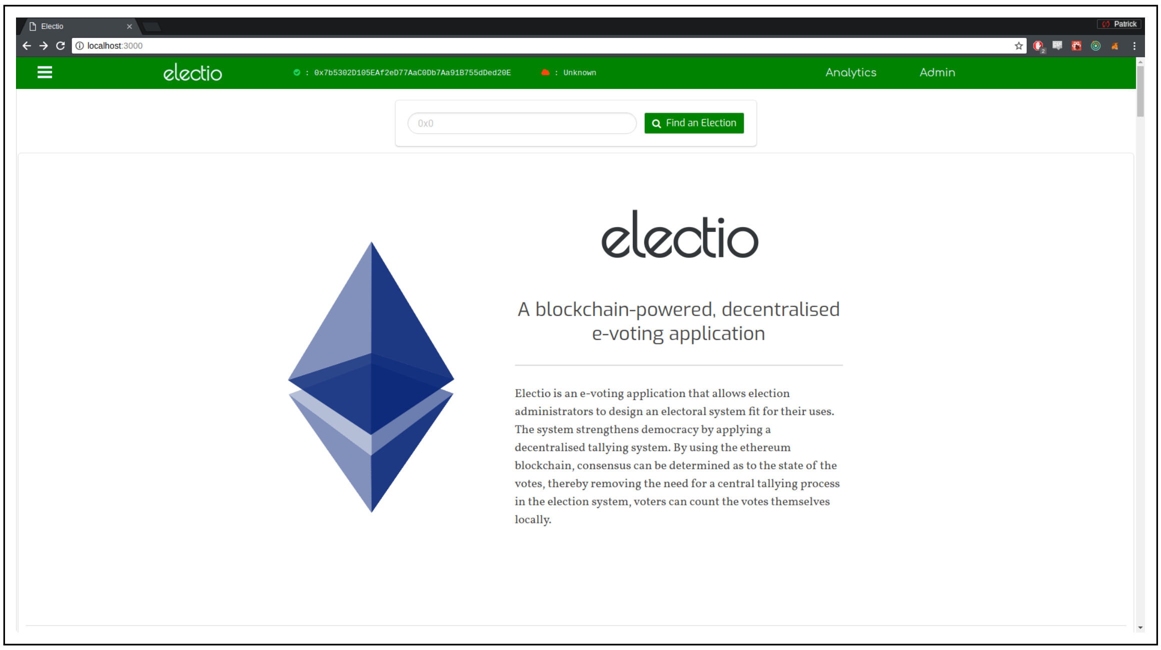
Authentication is exercised using the MetaMask plugin as before. If the screen above is displayed than MetaMask is not installed. The link to **Get MetaMask** will install the latest working version of MetaMask which will not work for this system. If the above is the case, please revert back to the manual instruction.



We see your MetaMask account is locked, please unlock it to continue

Where MetaMask is locked, the above will be displayed.

5.2 Landing Page



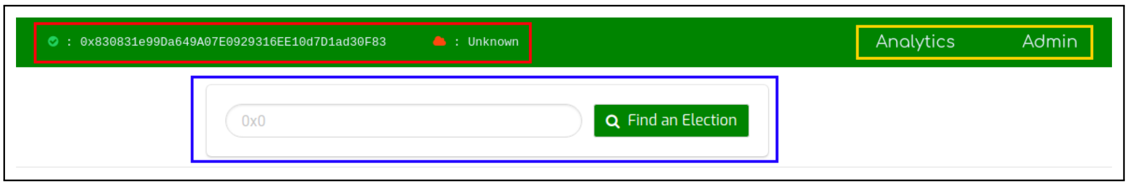
✓ : 0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E

: Unknown

A blockchain-powered, decentralised e-voting application

Electio is an e-voting application that allows election administrators to design an electoral system fit for their uses. The system strengthens democracy by applying a decentralised tallying system. By using the ethereum blockchain, consensus can be determined as to the state of the votes, thereby removing the need for a central tallying process in the election system, voters can count the votes themselves locally.

This is the landing page for Electio which is presented to the user once MetaMask is unlocked. The landing page has no function in the system other than explaining the core concept and ethos of the project.



✓ : 0x830831e99Da649A07E0929316EE10d7D1ad30F83

: Unknown

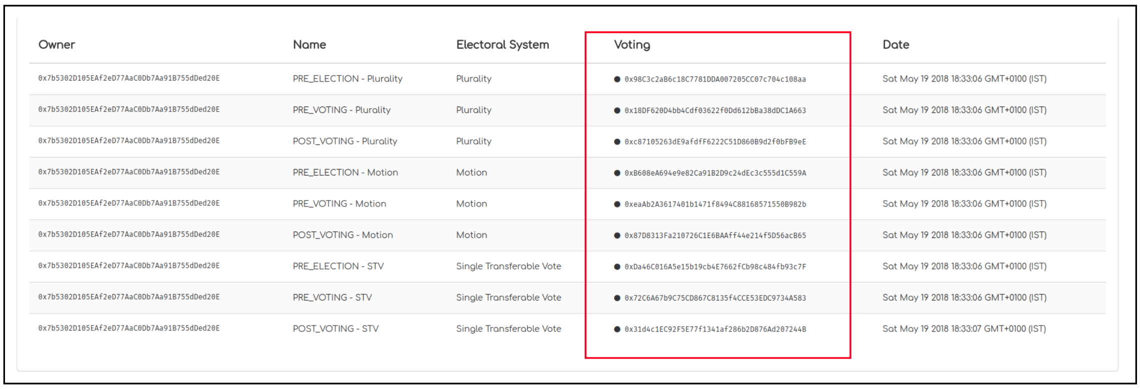
The main things to note on the landing page are uniform across the application.

- The wallet information labelled in red displays the current user's ethereum account address and the network they are connected to. This changes dynamically in response to the user changing network or address. Unkown represents any custom network.
- In yellow is the navigation links to the Analytics and Admin pages. The Analytics page merely lists all elections created on the system and enables any

user to find any election. The Admin page enables users to become the administrator of an election which will be explained later.

- In blue is the search bar. An election is represented with an ethereum address similar to the one which represents a user account and the deployer address. This election address is the location of the voting contract of a created election. By copying and pasting that address into the search input, that election will be loaded.

5.3 Analytics Page

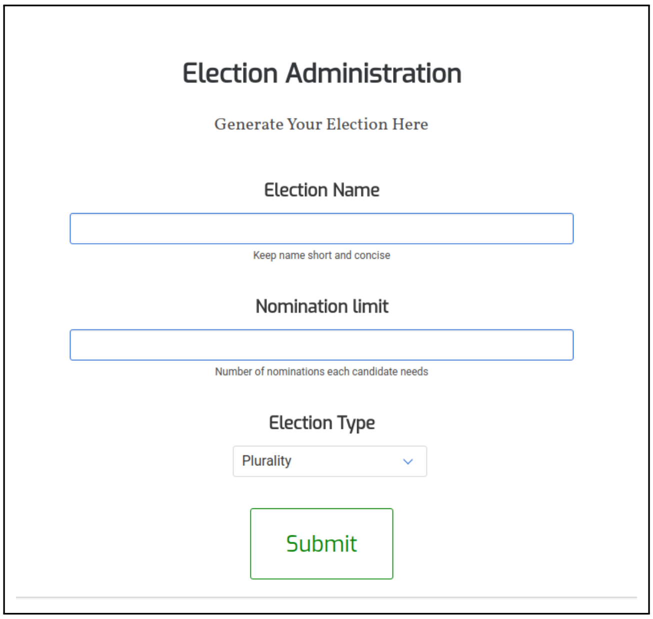


Owner	Name	Electoral System	Voting	Date
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	PRE_ELECTION - Plurality	Plurality	● 0x98C3c2aB6c18C7781DDA007205CC07c704c108aa	Sat May 19 2018 18:33:06 GMT+0100 (IST)
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	PRE_VOTING - Plurality	Plurality	● 0x18DF620D4bb4Cdf03622f0Dd612bBa38dDC1A663	Sat May 19 2018 18:33:06 GMT+0100 (IST)
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	POST_VOTING - Plurality	Plurality	● 0xc87105263dE9afdfF6222C51D860B9d2f0bFB9eE	Sat May 19 2018 18:33:06 GMT+0100 (IST)
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	PRE_ELECTION - Motion	Motion	● 0xB608eA694e9e82Ca91B2D9c24dEc3c555d1C559A	Sat May 19 2018 18:33:06 GMT+0100 (IST)
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	PRE_VOTING - Motion	Motion	● 0xeaAb2A3617401b1471f8494C88168571550B982b	Sat May 19 2018 18:33:06 GMT+0100 (IST)
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	POST_VOTING - Motion	Motion	● 0x87D8313Fa210726C1E6BAAff44e214f5D56acB65	Sat May 19 2018 18:33:06 GMT+0100 (IST)
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	PRE_ELECTION - STV	Single Transferable Vote	● 0xDa46C016A5e15b19cb4E7662fCb98c484fb93c7F	Sat May 19 2018 18:33:06 GMT+0100 (IST)
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	PRE_VOTING - STV	Single Transferable Vote	● 0x72C6A67b9C75CD867C8135f4CCE53EDC9734A583	Sat May 19 2018 18:33:06 GMT+0100 (IST)
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	POST_VOTING - STV	Single Transferable Vote	● 0x31d4c1EC92F5E77f1341af286b2D876Ad207244B	Sat May 19 2018 18:33:07 GMT+0100 (IST)

The above is the table display of the genScenario's script which produces 9 elections across different stages.

- The Owner column refers to the administrator of the election
- The Name column refers to the name of the election as instantiated by the administrator
- The Electoral System column refers to what election type that election is out of the three provided.
- The Voting column is the election address for each election, these are listed as links which if clicked loads that election as with the search bar option.
- The Date refers to the creation date of that election.

5.4 Admin Page - Election Creation



Election Administration

Generate Your Election Here

Election Name

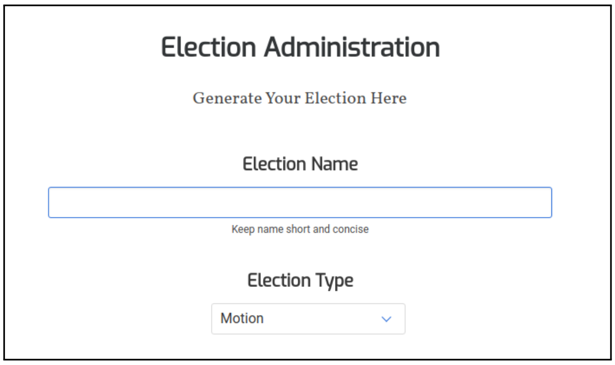
Keep name short and concise

Nomination limit

Number of nominations each candidate needs

Election Type: Plurality

The Admin page displays a simple form which can be used by any user of the system to create an election. The dropdown **Election Type** option will dynamically change specific to what is selected. In the case of plurality the above is represented. The **Nomination Limit** is of special significance for the election process as it defines the number of nominations required for a user who has applied to become a candidate to be validated.



Election Administration

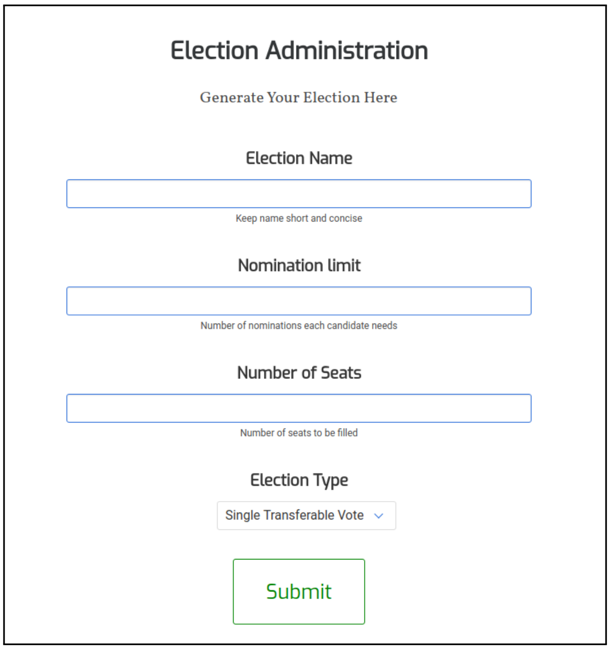
Generate Your Election Here

Election Name

Keep name short and concise

Election Type: Motion

The motion option does not require the nomination limit to be set as no human candidates utilise are to participate. Voters will vote on a motion which are defined by the administrator later on in the election.



Election Administration

Generate Your Election Here

Election Name

Keep name short and concise

Nomination limit

Number of nominations each candidate needs

Number of Seats

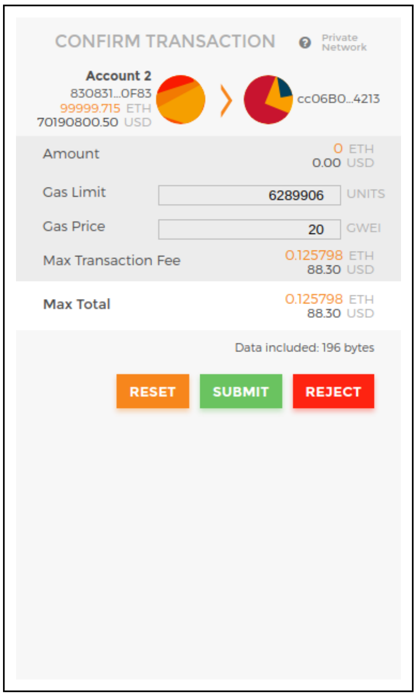
Number of seats to be filled

Election Type

Single Transferable Vote

The Single Transferable Vote option displays a **Number of Seats** input. This electoral system is for an election which allows users to elect a descending preference candidates to a number of seats.

5.5 Making a Transaction



CONFIRM TRANSACTION

Private Network

Account 2
830831...0F83
99999.715 ETH
70190800.50 USD

cc06B0...4213

Amount: 0 ETH, 0.00 USD

Gas Limit: 6289906 UNITS

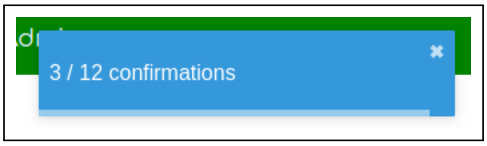
Gas Price: 20 GWEI

Max Transaction Fee: 0.125798 ETH, 88.30 USD

Max Total: 0.125798 ETH, 88.30 USD

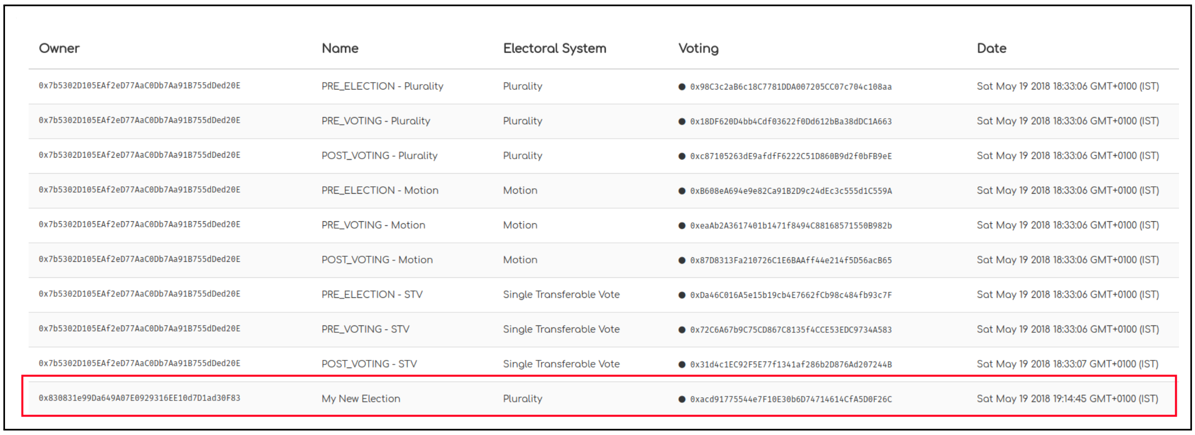
Data included: 196 bytes

Filling out a form with details of your choice and hitting submit will generate a popup like above. This popup is from MetaMask asking for confirmation for a transaction to be sent to the blockchain. This transaction contains data which is used to perform the logic on the smart contracts which generates an election as per the form details. By clicking submit, the transaction is sent to the blockchain awaiting to be accepted and mined.



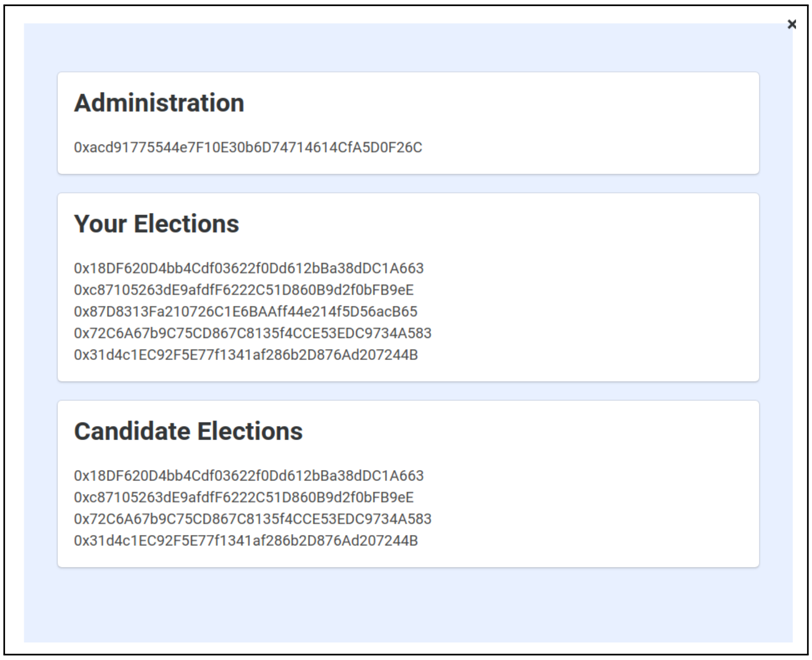
3 / 12 confirmations

Once submitted, a series of confirmation callbacks are received by the application. The 12 confirmations verifies that the transaction has succeeded.



Owner	Name	Electoral System	Voting	Date
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	PRE_ELECTION - Plurality	Plurality	● 0x98C3c2aB6c18C7781DDA007205CC07c704c108aa	Sat May 19 2018 18:33:06 GMT+0100 (IST)
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	PRE_VOTING - Plurality	Plurality	● 0x18DF620D4bb4Cdf03622f0Dd612bBa38dDC1A663	Sat May 19 2018 18:33:06 GMT+0100 (IST)
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	POST_VOTING - Plurality	Plurality	● 0xc87105263dE9afdfF6222C51D860B9d2f0bFB9eE	Sat May 19 2018 18:33:06 GMT+0100 (IST)
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	PRE_ELECTION - Motion	Motion	● 0xB608eA694e9e82Ca91B2D9c24dEc3c555d1C559A	Sat May 19 2018 18:33:06 GMT+0100 (IST)
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	PRE_VOTING - Motion	Motion	● 0xeaAb2A3617401b1471f8494C88168571550B982b	Sat May 19 2018 18:33:06 GMT+0100 (IST)
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	POST_VOTING - Motion	Motion	● 0x87D8313Fa210726C1E6BAAff44e214f5D56acB65	Sat May 19 2018 18:33:06 GMT+0100 (IST)
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	PRE_ELECTION - STV	Single Transferable Vote	● 0xDa46C016A5e15b19cb4E7662fCb98c484fb93c7F	Sat May 19 2018 18:33:06 GMT+0100 (IST)
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	PRE_VOTING - STV	Single Transferable Vote	● 0x72C6A67b9C75CD867C8135f4CCE53EDC9734A583	Sat May 19 2018 18:33:06 GMT+0100 (IST)
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	POST_VOTING - STV	Single Transferable Vote	● 0x31d4c1EC92F5E77f1341af286b2D876Ad207244B	Sat May 19 2018 18:33:07 GMT+0100 (IST)
0x830831e99Da649A07E0929316EE10d7D1ad30F83	My New Election	Plurality	● 0xacd91775544e7F10E30b6D74714614CfA5D0F26C	Sat May 19 2018 19:14:45 GMT+0100 (IST)

Navigating back to the Analytics page we find that the new election has been generated.



Administration

0xacd91775544e7F10E30b6D74714614CfA5D0F26C

Your Elections

0x18DF620D4bb4Cdf03622f0Dd612bBa38dDC1A663
0xc87105263dE9afdfF6222C51D860B9d2f0bFB9eE
0x87D8313Fa210726C1E6BAAff44e214f5D56acB65
0x72C6A67b9C75CD867C8135f4CCE53EDC9734A583
0x31d4c1EC92F5E77f1341af286b2D876Ad207244B

Candidate Elections

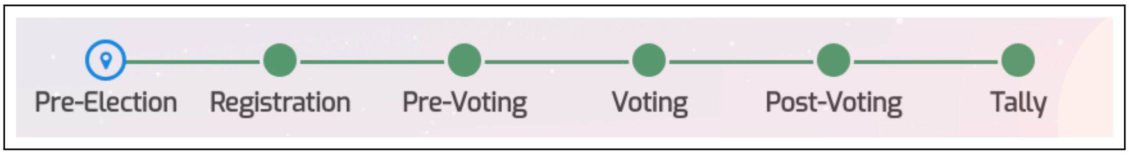
0x18DF620D4bb4Cdf03622f0Dd612bBa38dDC1A663
0xc87105263dE9afdfF6222C51D860B9d2f0bFB9eE
0x72C6A67b9C75CD867C8135f4CCE53EDC9734A583
0x31d4c1EC92F5E77f1341af286b2D876Ad207244B

The hamburger menu which exists on the extreme left of the top navigation bar produces a slideout list of the elections specific to a user.

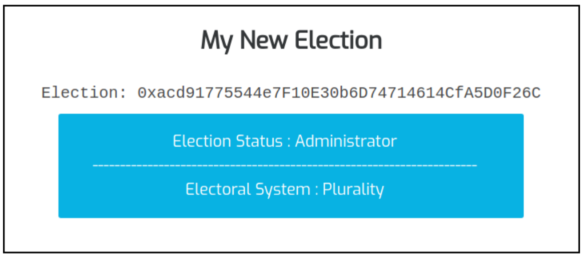
- Under **Administration** is all the elections in which the user has created
- Under **Your Elections** is all elections the user is a participant voter
- Under **Candidate Elections** is all elections the user is a candidate

6 Election Stages

All state-level elections are constructed within a timeline which I have designed the application around. This timeline is broken into a number of stages which enable the system to designate a specific set of actions to certain users of the election suitable to that stage.



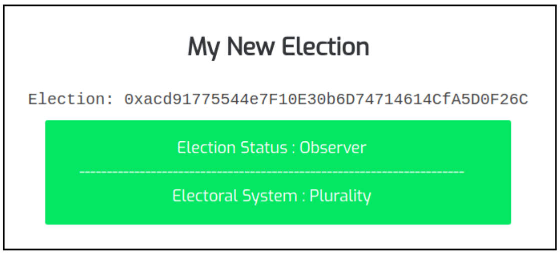
The above is the election timeline which is displayed during every election. The current stage marker progresses left-to-right across the timeline as the election occurs. The 9 scenario's generated by the script sets the 3 electoral systems in Pre-Election, Pre-Voting and Post-Voting stages. Both the Registration and Voting stages are time-windowed and the Tally stage will be a permanent view of the election when the election is over.


My New Election

Election: 0xacd91775544e7F10E30b6D74714614CfA5D0F26C

Election Status : Administrator

Electoral System : Plurality


My New Election

Election: 0xacd91775544e7F10E30b6D74714614CfA5D0F26C

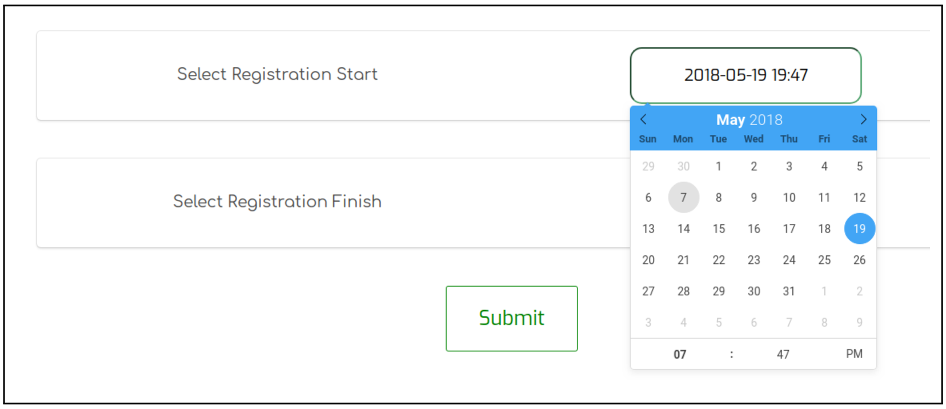
Election Status : Observer

Electoral System : Plurality

The above is a representation of the election status which changes dynamically to what user is signed in to MetaMask.

6.1 Pre-Election

Before an election starts all users with the exception of the Administrator are Observers. Observers are all outside participants in the election and have a limited role in the process.



Select Registration Start: 2018-05-19 19:47

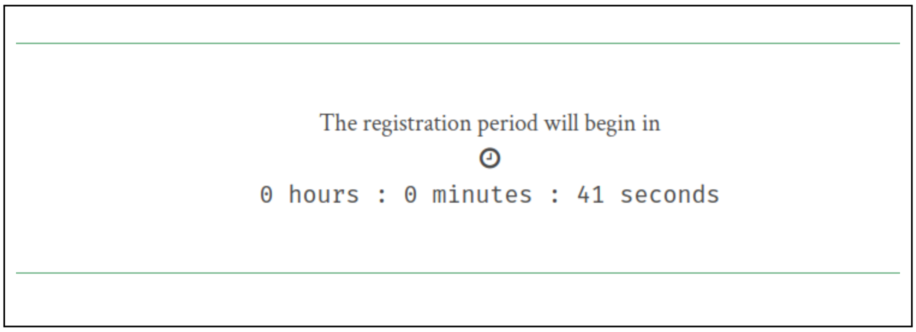
Select Registration Finish

May 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

07 : 47 PM

The Administrator will be displayed two datepicker calendars which are used to select a time-window in the future for the registration period. This time-window ought to be long enough so that all particpants can register at their convenience. Submitting generates a MetaMask popup like with election creation.



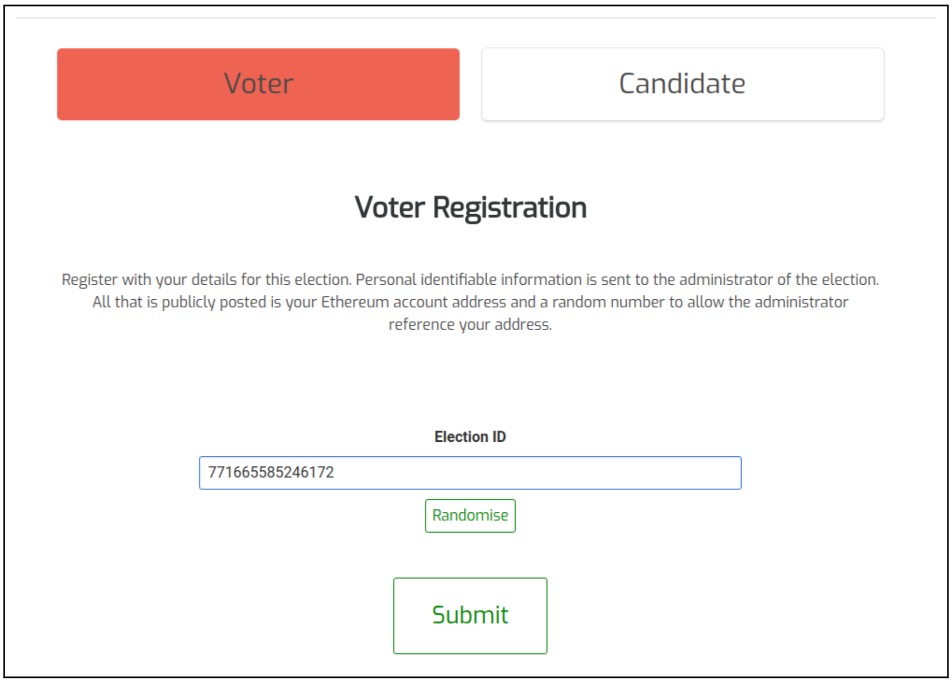
The registration period will begin in

0 hours : 0 minutes : 41 seconds

When the registration time-window is set, a countdown timer for when that begins will be displayed.

6.2 Registration - Plurality & STV

6.2.1 Voter

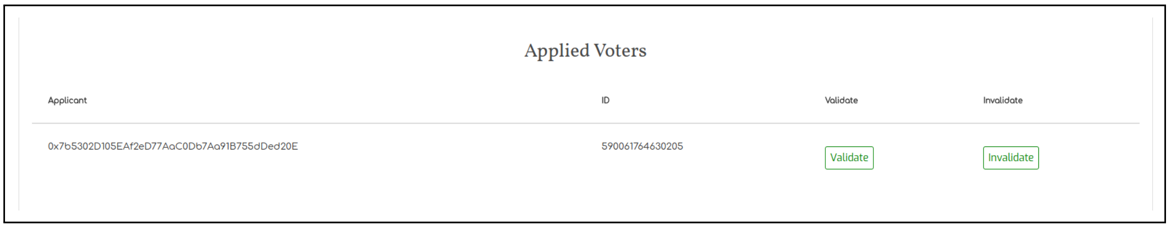


Voter Registration

Register with your details for this election. Personal identifiable information is sent to the administrator of the election. All that is publicly posted is your Ethereum account address and a random number to allow the administrator reference your address.

Election ID: 771665585246172

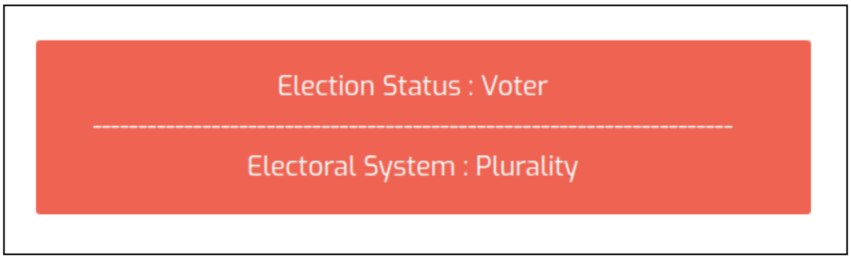
In the registration stage, all observers are able to apply as voters and candidates. To register as a voter, the observer submits a random Election ID to the blockchain. This Election ID is to be sent outside of this application along with identification information to the Administrator. If the Administrator identifies an applicant as a valid voter, they can correspond the Election ID they were given with the one mapped to the applicants address on the blockchain.



Applied Voters

Applicant	ID	Validate	Invalidate
0x7b5302D105EAf2eD77AaC0Db7Aa91B755dDed20E	590061764630205	Validate	Invalidate

The Administrator view displays a table of data which fills as new observers apply. Administrators can choose to validate or invalidate users. Once validated by the Administrator, that address is identified by the system as a Voter.

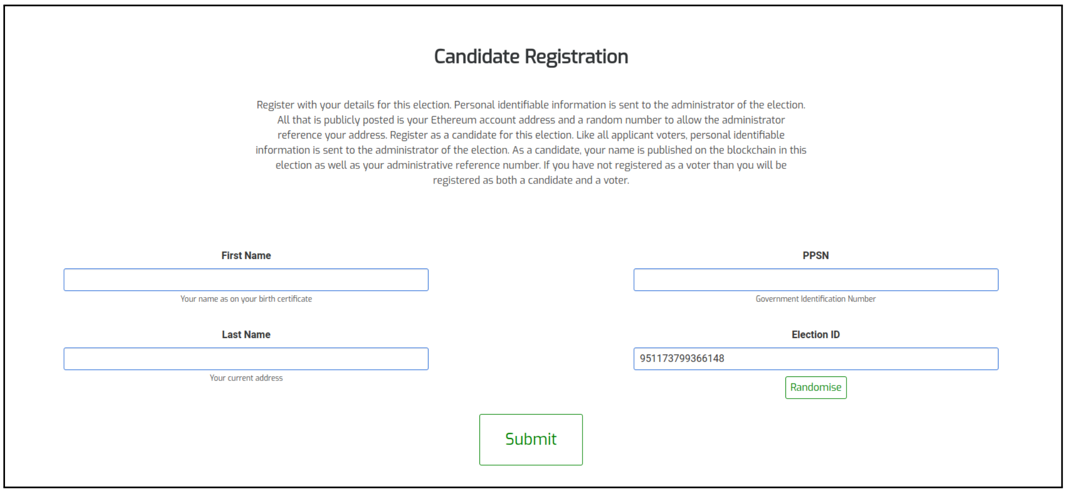


Election Status : Voter

Electoral System : Plurality

Voters will then have an election status as above.

6.2.2 Candidate



Candidate Registration

Register with your details for this election. Personal identifiable information is sent to the administrator of the election. All that is publicly posted is your Ethereum account address and a random number to allow the administrator reference your address. Register as a candidate for this election. Like all applicant voters, personal identifiable information is sent to the administrator of the election. As a candidate, your name is published on the blockchain in this election as well as your administrative reference number. If you have not registered as a voter than you will be registered as both a candidate and a voter.

First Name

Your name as on your birth certificate

Last Name

Your current address

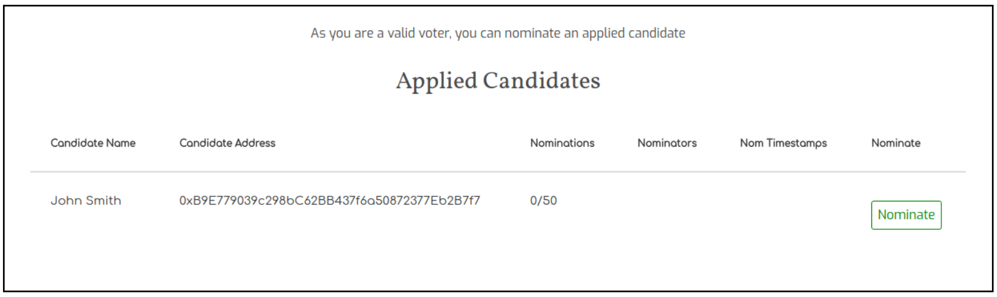
PPSN

Government Identification Number

Election ID

951173799366148

Registration as a Candidate is also straightforward as users submit their Election ID as a Voter would but also submit their first and last name to the blockchain. Candidates are not deemed valid by the Administrator, but instead are validated by the Voters.

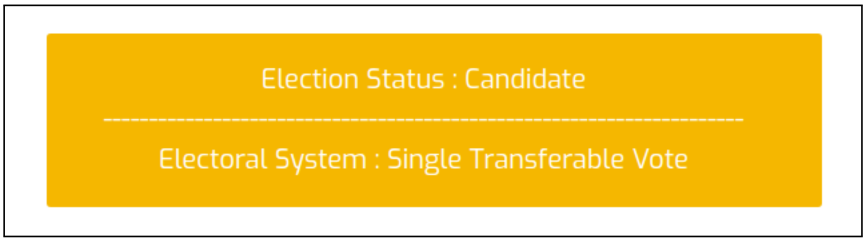


As you are a valid voter, you can nominate an applied candidate

Applied Candidates

Candidate Name	Candidate Address	Nominations	Nominators	Nom Timestamps	Nominate
John Smith	0xB9E779039c298bC62BB437f6a50872377Eb2B7f7	0/50			Nominate

The nomination limit set by the Administrator is illustrated here as the number of nominations required by the applicant Candidate. This is the view displayed to a Voter during the registration period. Each voter has only one nomination and can nominate a candidate by clicking the **Nominate** button. When a candidate is nominated, the address of the nominator and the time of nomination is recorded.

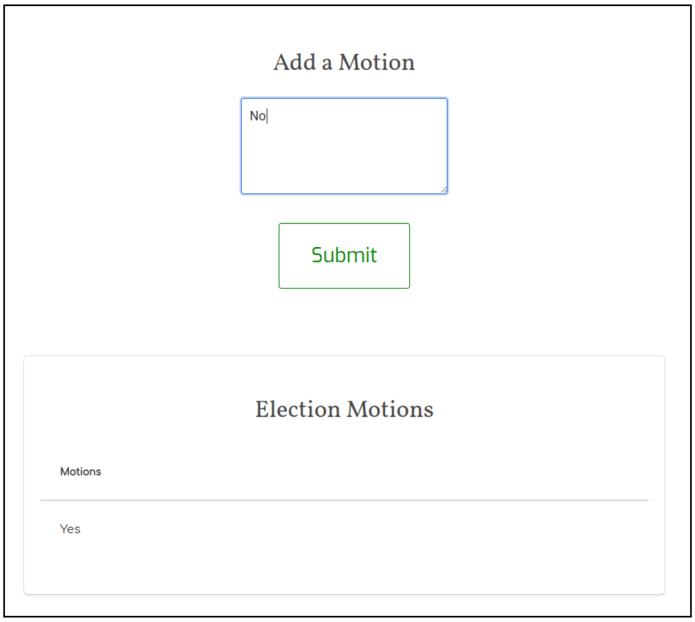

Election Status : Candidate

Electoral System : Single Transferable Vote

When an Observer receieves enough nominations, they are then validate. Their election status will also change to the above.

6.3 Registration - Motion

The motion electoral system differs from the Candidate registration stage. In this section, an Observer can only register as a Voter and the Administrator selects the motions which are to be voted on.



Add a Motion

No

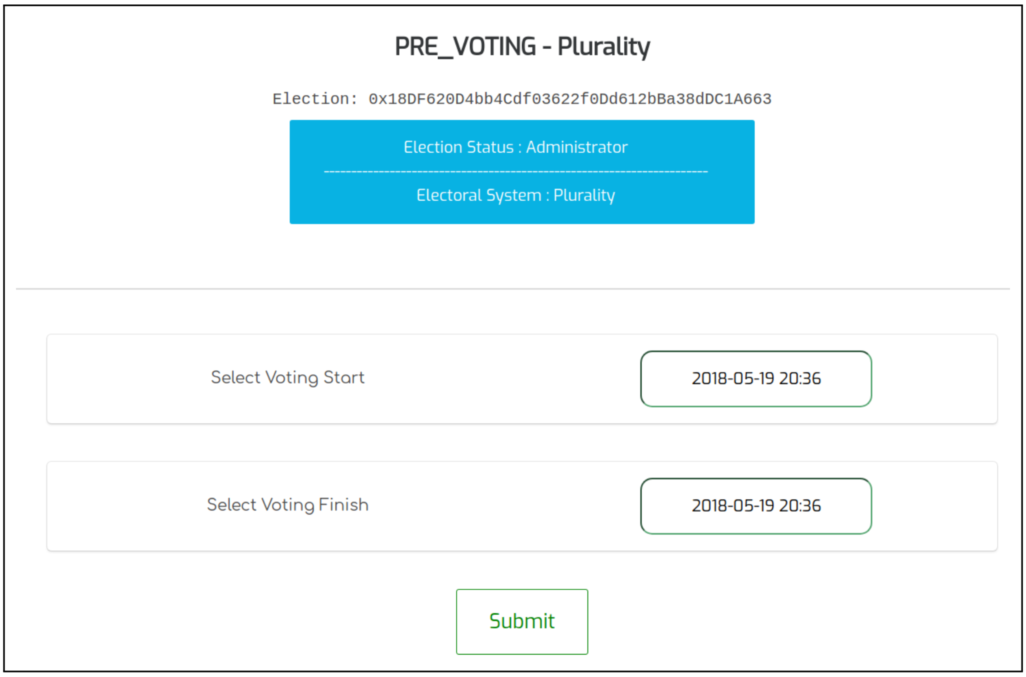
Election Motions

Motions
Yes

The Administrator in this case simply fills an input box with the motion to be voted.

6.4 Pre-Voting

The pre-voting stage, much like the pre-election stage is about setting the voting time-window. Also a countdown timer will be displayed for when that time will be.



PRE_VOTING - Plurality

Election: 0x18DF620D4bb4Cdf03622f0Dd612bBa38dDC1A663

Election Status : Administrator

Electoral System : Plurality

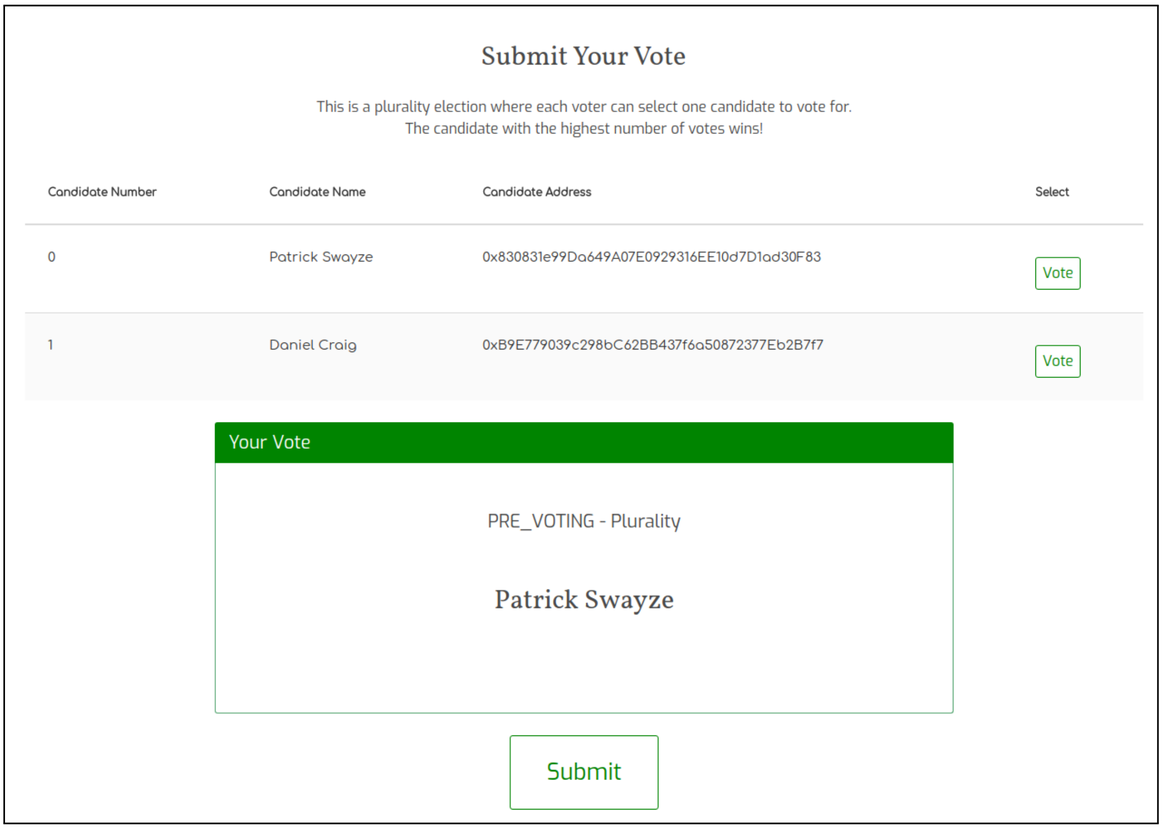
Select Voting Start 2018-05-19 20:36

Select Voting Finish 2018-05-19 20:36

In the background to this, the Node server will generate a public-private keypair which will be used to encrypt and tally the votes later on. The public key is published on the blockchain during this stage.

6.5 Voting - Plurality & Motion

Voting in the plurality system is straightforward and involves a Voter to select a single favoured candidate. Users who registered as Candidates are also registered as Voters should the Administrator deem them valid.



Submit Your Vote

This is a plurality election where each voter can select one candidate to vote for.
The candidate with the highest number of votes wins!

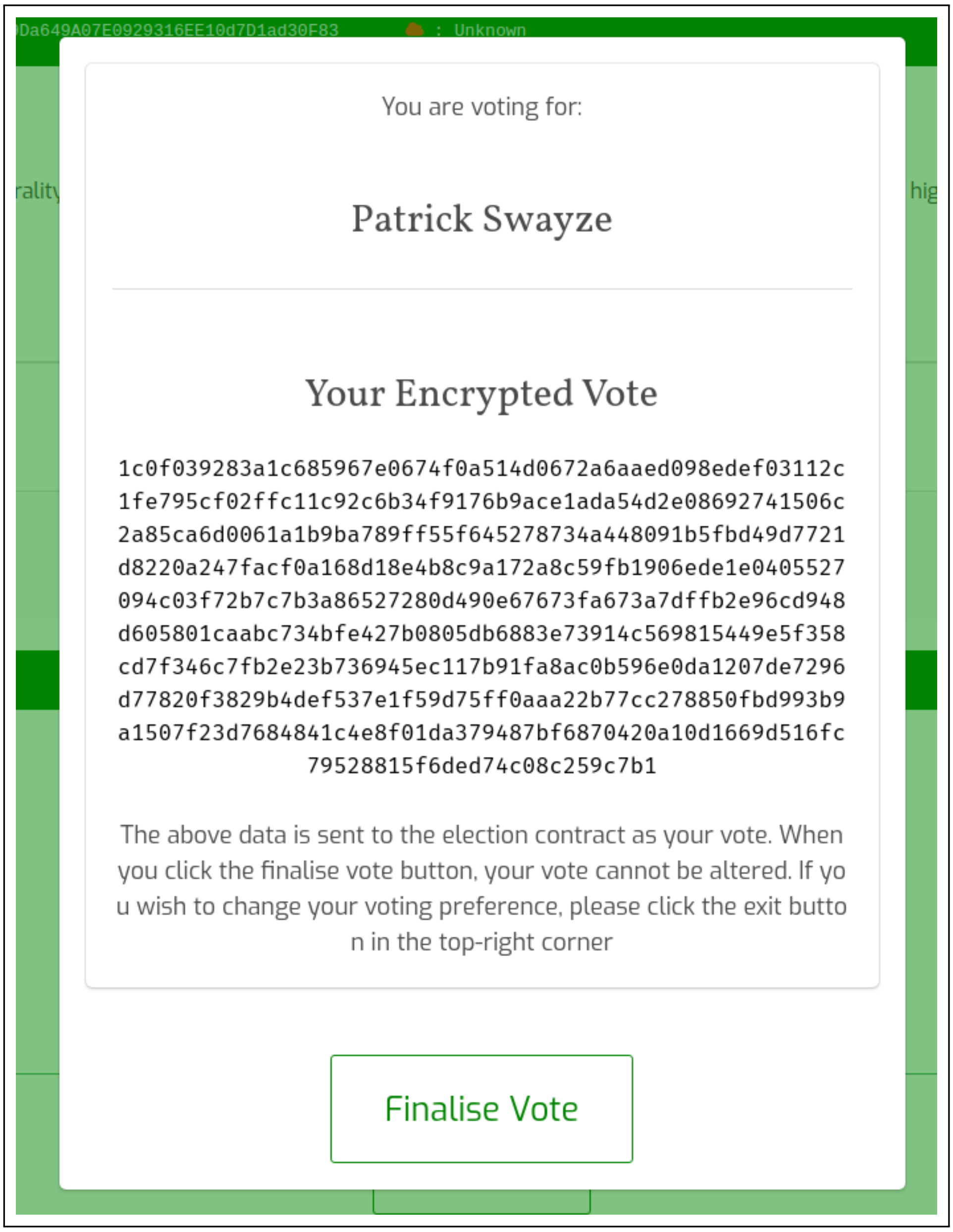
Candidate Number	Candidate Name	Candidate Address	Select
0	Patrick Swayze	0x830831e99Da649A07E0929316EE10d7D1ad30F83	Vote
1	Daniel Craig	0xB9E779039c298bC62BB437f6a50872377Eb2B7f7	Vote

Your Vote

PRE_VOTING - Plurality

Patrick Swayze

The **Your Vote** box indicates the selected vote and will change on the selection of any other candidate. Selecting submit will present the Voter with a final chance to change their vote.



You are voting for:

Patrick Swayze

Your Encrypted Vote

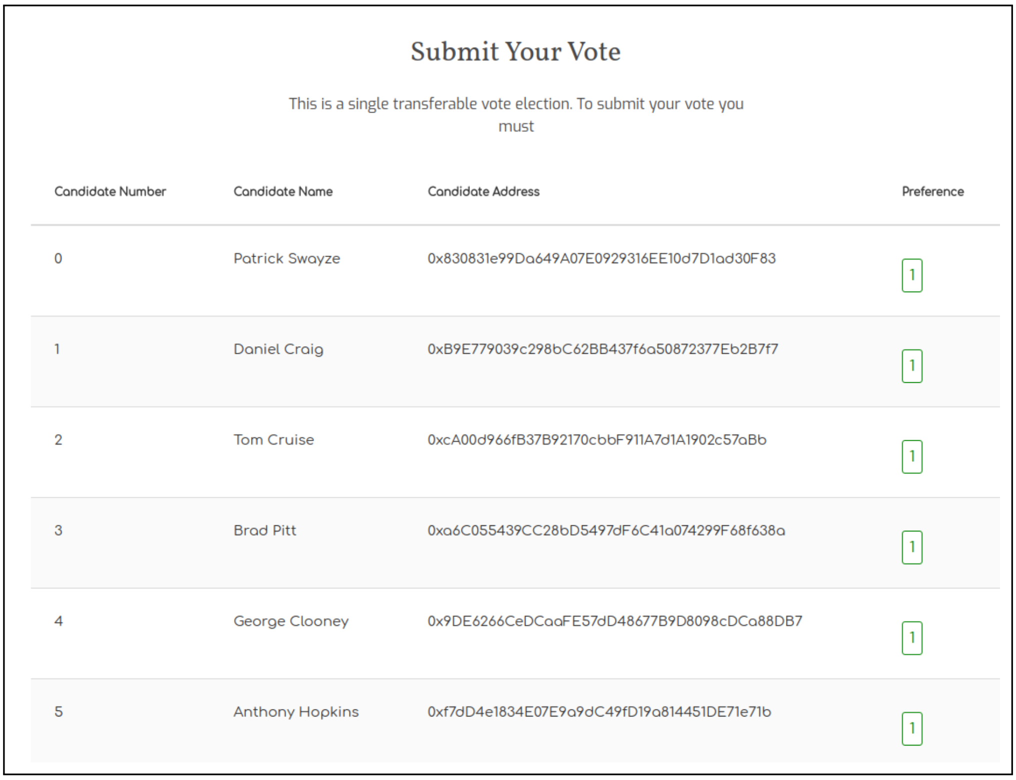
```
1c0f039283a1c685967e0674f0a514d0672a6aaed098edef03112c
1fe795cf02ffc11c92c6b34f9176b9ace1ada54d2e08692741506c
2a85ca6d0061a1b9ba789ff55f645278734a448091b5fbd49d7721
d8220a247facf0a168d18e4b8c9a172a8c59fb1906ede1e0405527
094c03f72b7c7b3a86527280d490e67673fa673a7dffb2e96cd948
d605801caabc734bfe427b0805db6883e73914c569815449e5f358
cd7f346c7fb2e23b736945ec117b91fa8ac0b596e0da1207de7296
d77820f3829b4def537e1f59d75ff0aaa22b77cc278850fbd993b9
a1507f23d7684841c4e8f01da379487bf6870420a10d1669d516fc
79528815f6ded74c08c259c7b1
```

The above data is sent to the election contract as your vote. When you click the finalise vote button, your vote cannot be altered. If you wish to change your voting preference, please click the exit button in the top-right corner

When a vote is to be finalised, the Voter is represented with a voting card containing their preference and their encrypted vote. This encryption is submitted to the blockchain and remains encrypted until all votes are gathered after the election. This method is also the same for the motion electoral system where one voting option is selected.

6.6 Voting - Single Transferable Vote

Voting in an STV election has a slight variation in comparison to the plurality system. Voters in STV elections are able to make their vote in the form of descending preference.

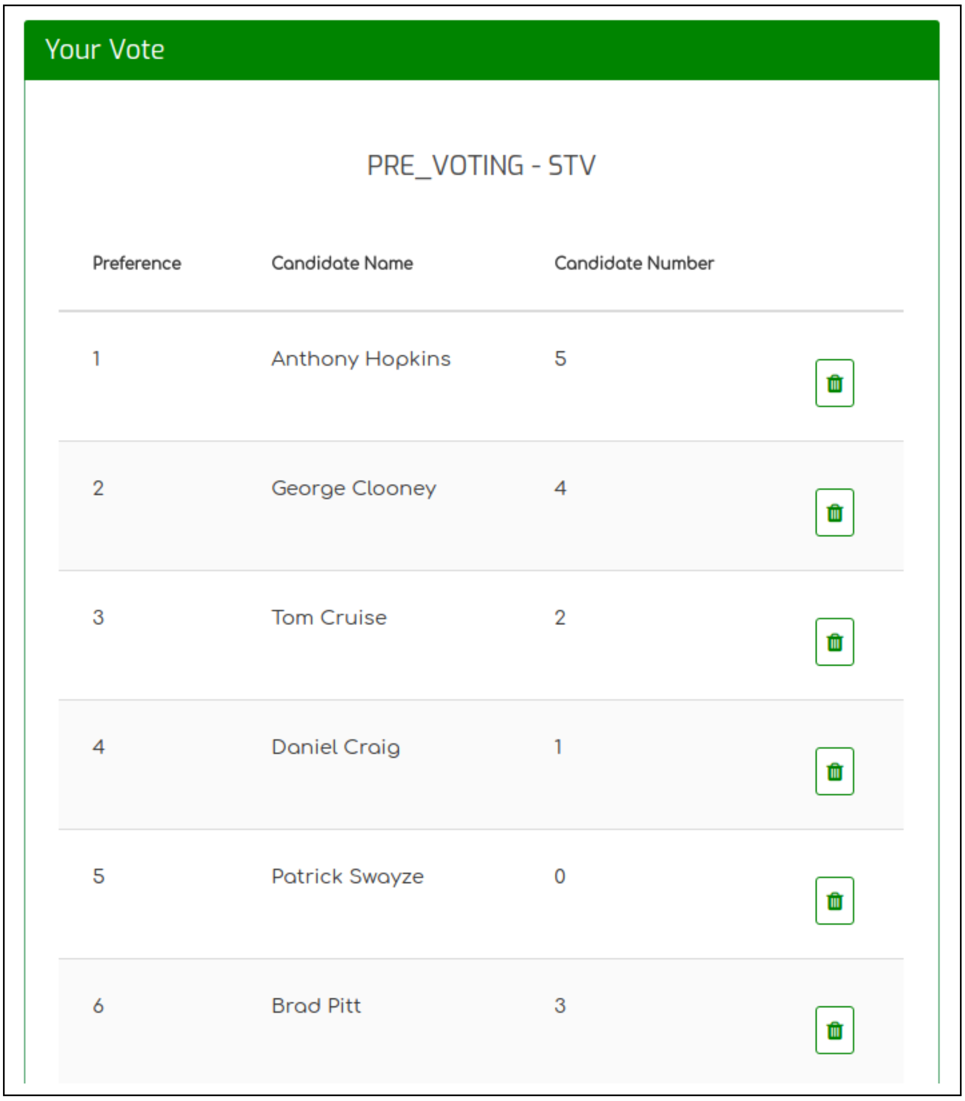


Submit Your Vote

This is a single transferable vote election. To submit your vote you must

Candidate Number	Candidate Name	Candidate Address	Preference
0	Patrick Swayze	0x830831e99Da649A07E0929316EE10d7D1ad30F83	1
1	Daniel Craig	0xB9E779039c298bC62BB437f6a50872377Eb2B7f7	1
2	Tom Cruise	0xcA00d966fB37B92170cbbF911A7d1A1902c57aBb	1
3	Brad Pitt	0xa6C055439CC28bD5497dF6C41a074299F68f638a	1
4	George Clooney	0x9DE6266CeDCaaFE57dD48677B9D8098cDCa88DB7	1
5	Anthony Hopkins	0xf7dD4e1834E07E9a9dC49fD19a814451DE71e71b	1

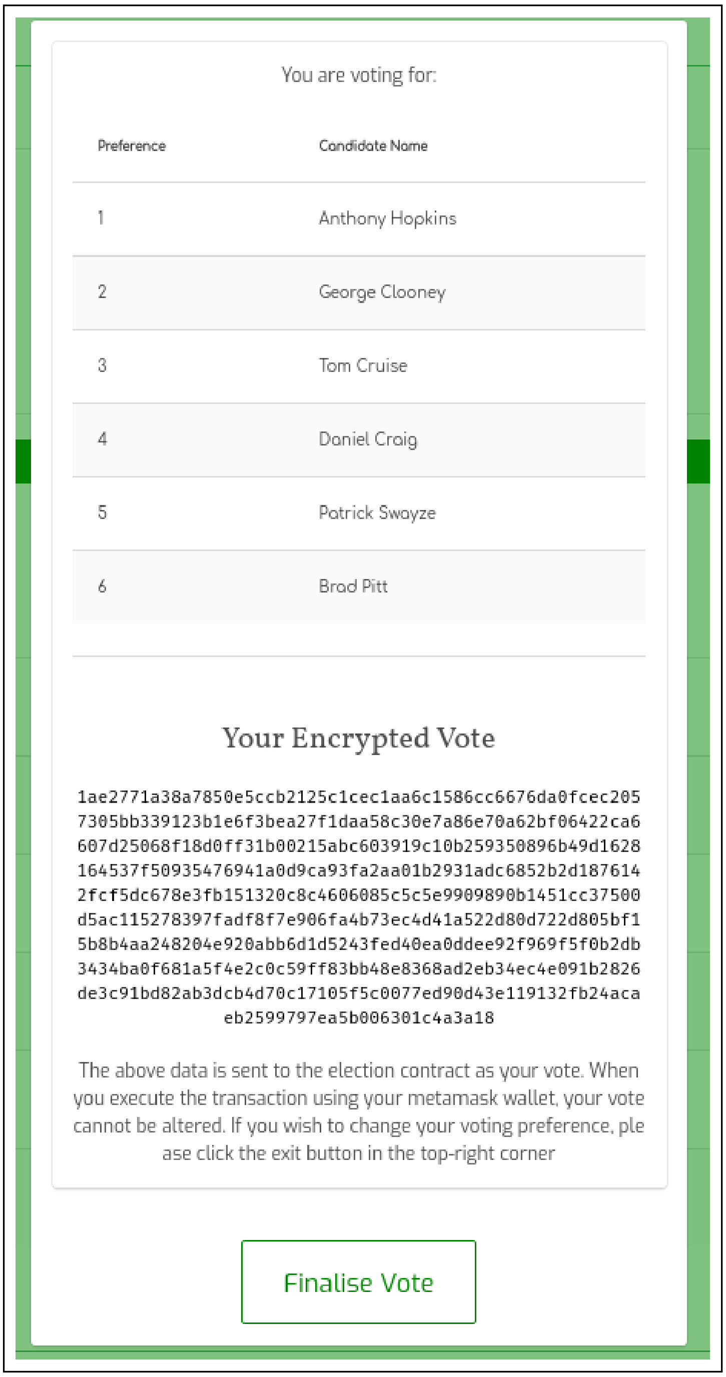
This table is what a Voter in a STV election would be displayed. The electio system records the votes as before in a **Your Vote** table where selections can be changed and removed. In order to vote, the rule is that the submitted vote must contain at least one preference.



PRE_VOTING - STV

Preference	Candidate Name	Candidate Number	
1	Anthony Hopkins	5	🗑
2	George Clooney	4	🗑
3	Tom Cruise	2	🗑
4	Daniel Craig	1	🗑
5	Patrick Swayze	0	🗑
6	Brad Pitt	3	🗑

In this instance above, a Voter could select all displayed candidates. Should a Voter wish to remove a preference, they would select the delete icon.



You are voting for:

Preference	Candidate Name
1	Anthony Hopkins
2	George Clooney
3	Tom Cruise
4	Daniel Craig
5	Patrick Swayze
6	Brad Pitt

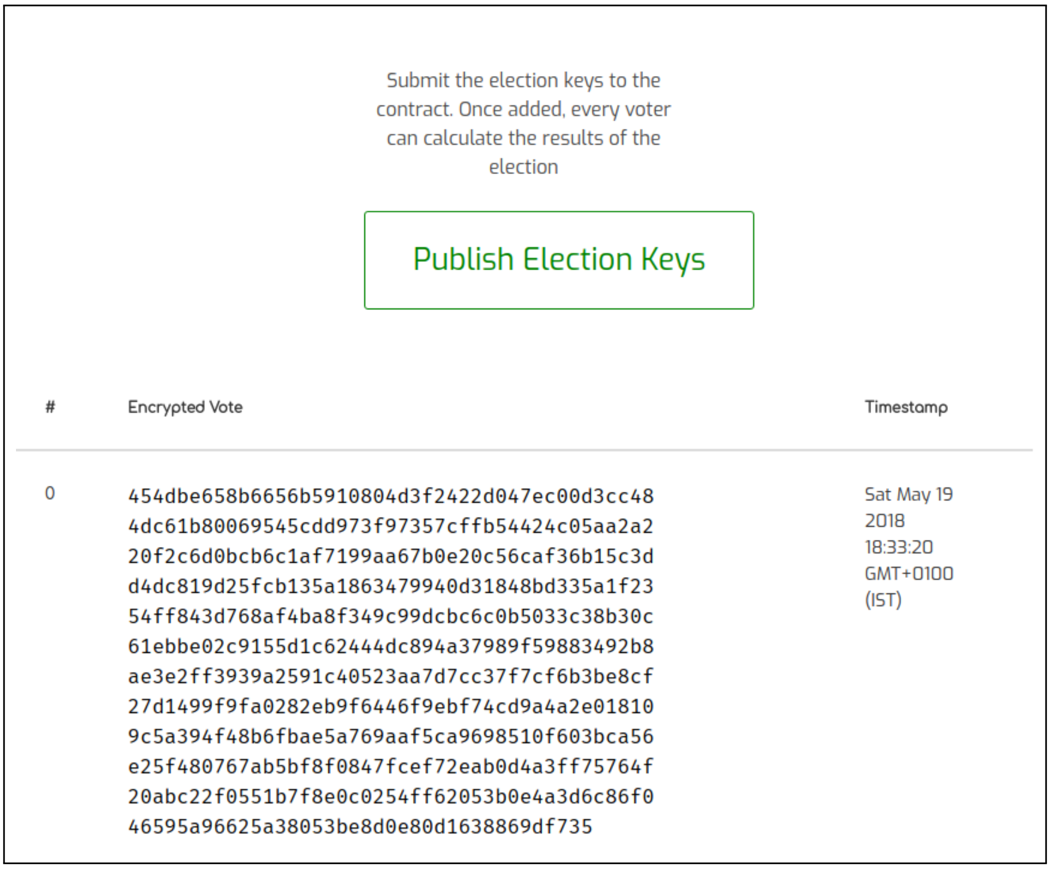
Your Encrypted Vote

```
1ae2771a38a7850e5ccb2125c1cec1aa6c1586cc6676da0fcec205
7305bb339123b1e6f3bea27f1daa58c30e7a86e70a62bf06422ca6
607d25068f18d0ff31b00215abc603919c10b259350896b49d1628
164537f50935476941a0d9ca93fa2aa01b2931adc6852b2d187614
2fcf5dc678e3fb151320c8c4606085c5c5e9909890b1451cc37500
d5ac115278397fadf8f7e906fa4b73ec4d41a522d80d722d805bf1
5b8b4aa248204e920abb6d1d5243fed40ea0ddee92f969f5f0b2db
3434ba0f681a5f4e2c0c59ff83bb48e8368ad2eb34ec4e091b2826
de3c91bd82ab3dcb4d70c17105f5c0077ed90d43e119132fb24aca
eb2599797ea5b006301c4a3a18
```

The above data is sent to the election contract as your vote. When you execute the transaction using your metamask wallet, your vote cannot be altered. If you wish to change your voting preference, ple ase click the exit button in the top-right corner

The finalise vote menu also generates the final selection for that Voter, giving them the chance to change if necessary. Once the transaction is submitted, the vote is final and cannot be changed.

6.7 Post-Voting



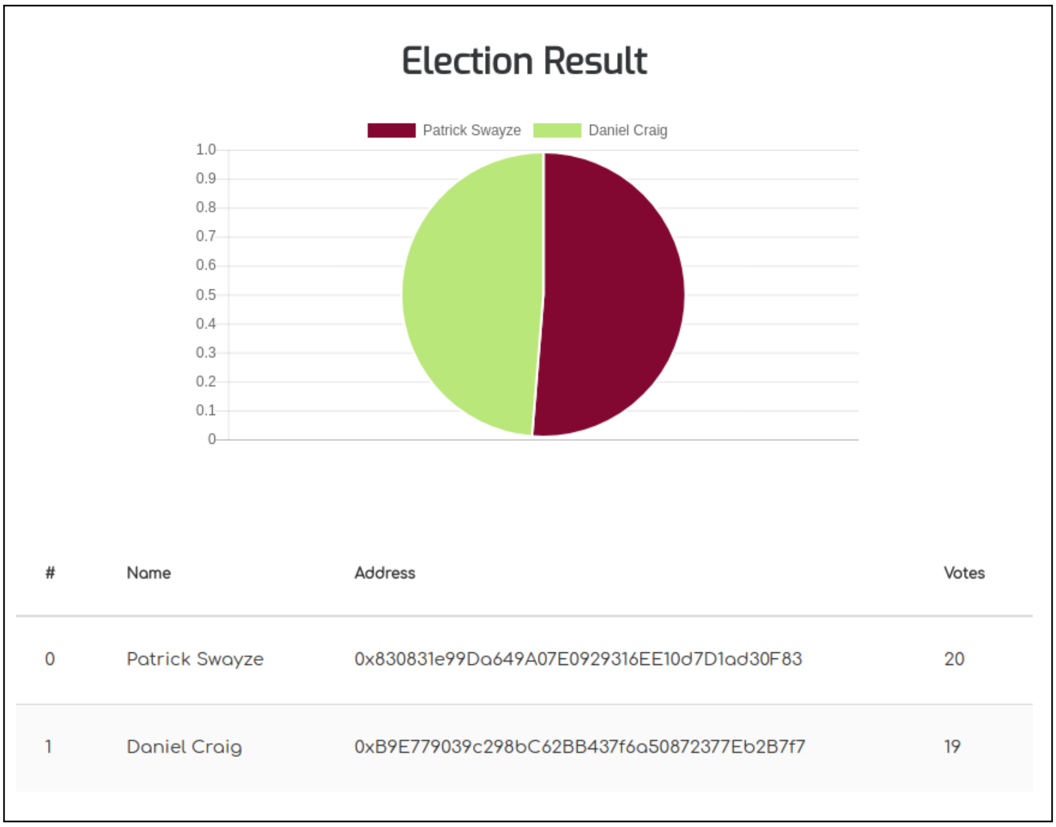
Submit the election keys to the contract. Once added, every voter can calculate the results of the election

#	Encrypted Vote	Timestamp
0	454dbe658b6656b5910804d3f2422d047ec00d3cc48 4dc61b80069545cdd973f97357cffb54424c05aa2a2 20f2c6d0bcb6c1af7199aa67b0e20c56caf36b15c3d d4dc819d25fcb135a1863479940d31848bd335a1f23 54ff843d768af4ba8f349c99dcbc6c0b5033c38b30c 61ebbe02c9155d1c62444dc894a37989f59883492b8 ae3e2ff3939a2591c40523aa7d7cc37f7cf6b3be8cf 27d1499f9fa0282eb9f6446f9ebf74cd9a4a2e01810 9c5a394f48b6fbae5a769aaf5ca9698510f603bca56 e25f480767ab5bf8f0847fcef72eab0d4a3ff75764f 20abc22f0551b7f8e0c0254ff62053b0e4a3d6c86f0 46595a96625a38053be8d0e80d1638869df735	Sat May 19 2018 18:33:20 GMT+0100 (IST)

The above is the post-voting stage for the Administrator. During this stage, all encrypted votes are displayed which gives Voters some meaningful feedback as to the particpation of the election. The Publish Election Keys button will publish the private-keys of the election to the blockchain. Only the Administrator can do this.These keys will be used by all users during the tallying process to generate an election result.

6.8 Tallying - Plurality & Motion

This is the final stage for all elections and is viewable by all users of the system. It enables anyone to search for any past election and regenerate the results of the election as time goes on. As the blockchain is permanent, the results are permanent and gives a degree of finality and assurance as to the result.



#	Name	Address	Votes
0	Patrick Swayze	0x830831e99Da649A07E0929316EE10d7D1ad30F83	20
1	Daniel Craig	0xB9E779039c298bC62BB437f6a50872377Eb2B7f7	19

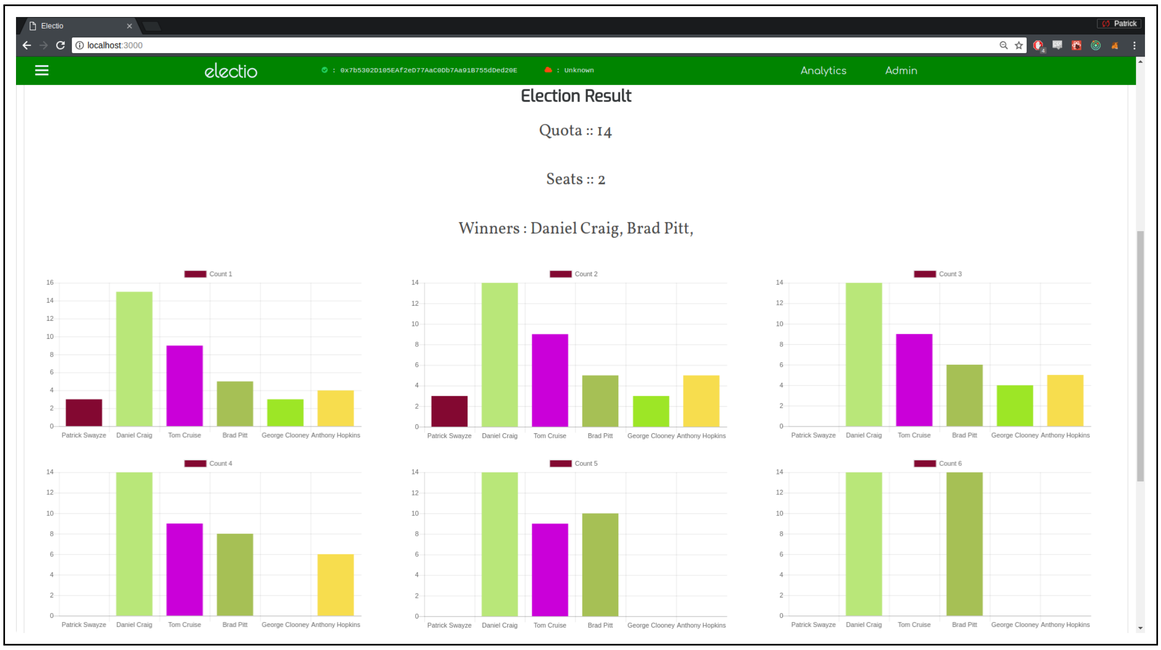
The result of this election is represented in a pie-chart and a table. These results are auto-generated on the loading of the election in the application. In both the plurality and motion voting systems, the result is straightforward and the table is presented in order of most votes.

6.9 Tallying - Single Transferable Vote

The results of the Single Transferable Vote system is more complicated as the result is generated by distributing all votes for any Candidate over the election quota or the removal and distribution of votes of the lowest remaining candidate. One pass of either of these actions is called a count. The election quota, is calculated by :

```
Math.floor(num_voters / (num_seats + 1) + 1)
```

When either the number of candidates that remain equal the number of seats to be filled or the number of candidates that have reached the quota equal the seats to be filled, those candidates are elected.



Election Result

Quota :: 14

Seats :: 2

Winners : Daniel Craig, Brad Pitt,

The result of an STV election is displayed as above for all Electio users. Each bar chart represents a count and at each stage a user can visualise the process. In the final slide, the two remaining candidates have reached the quota and are deemed elected.