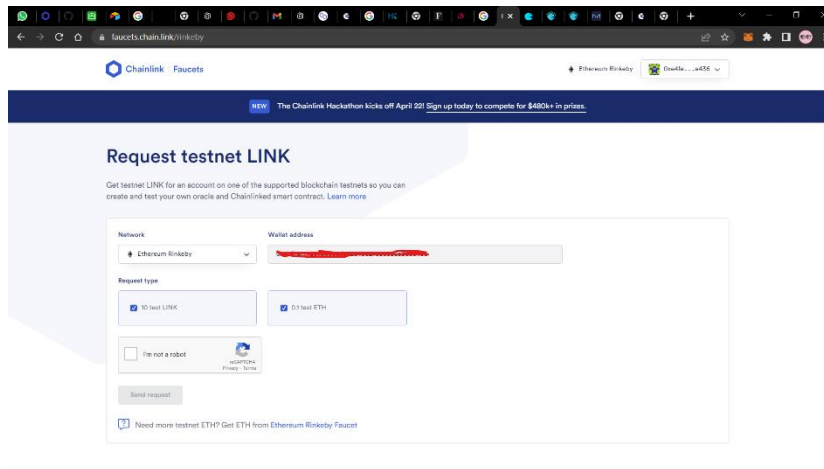


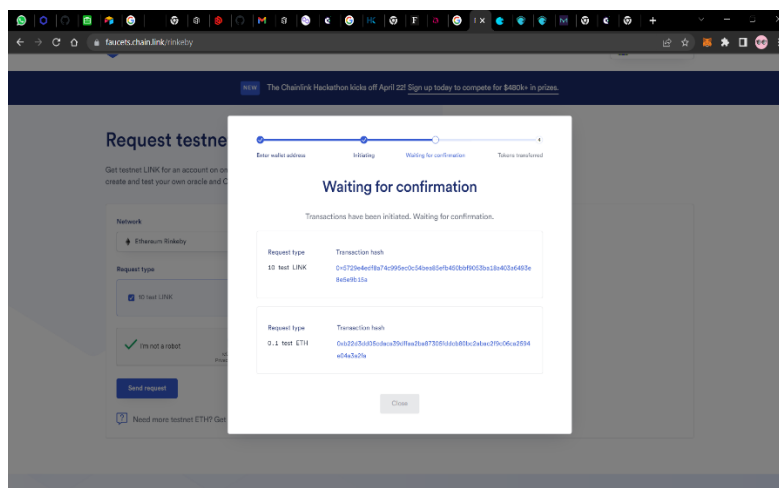
# CREATING A TEST ETH

- Deepthi Rajakumar

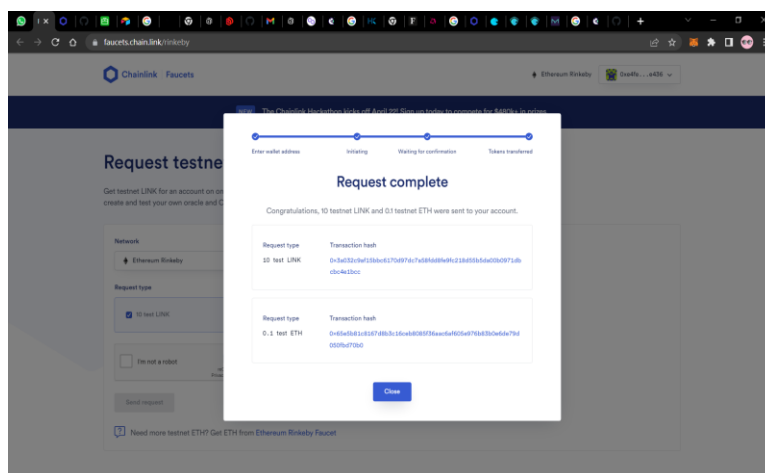
1. Log in to <https://faucets.chain.link/rinkeby>
2. Connect your meta mask account with the Rinkeby faucet.



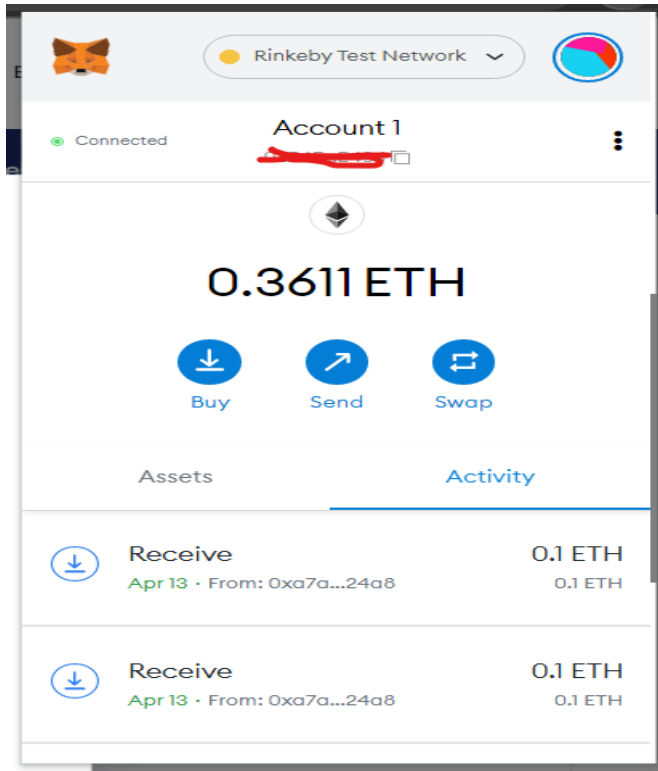
3. Click on Send request.



- 4.



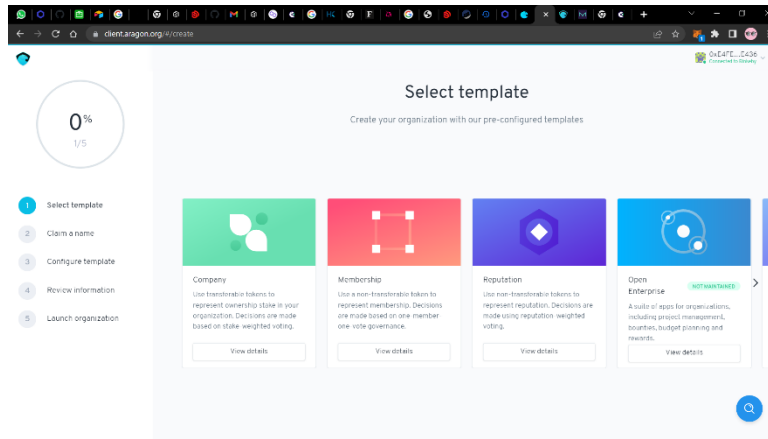
5. Now on our wallet we can notice an increase in our etherium.



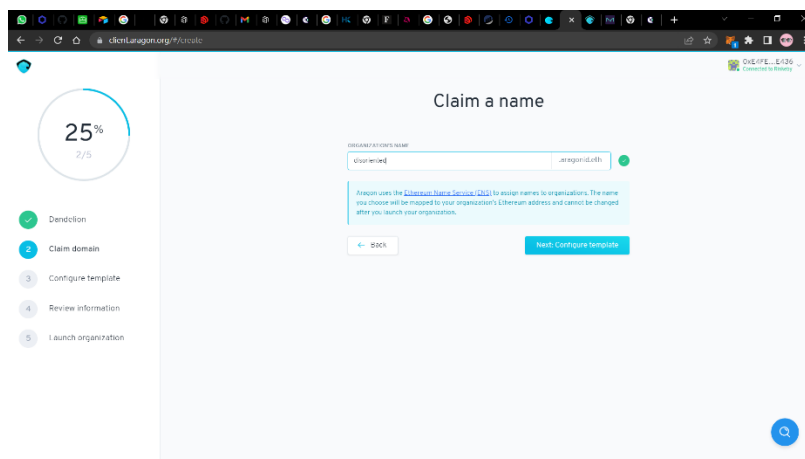
# CREATING YOUR OWN DAO

## -Decentralised autonomous organization

1. Open Aragon and connect your meta mask wallet.



2. Choose a template.



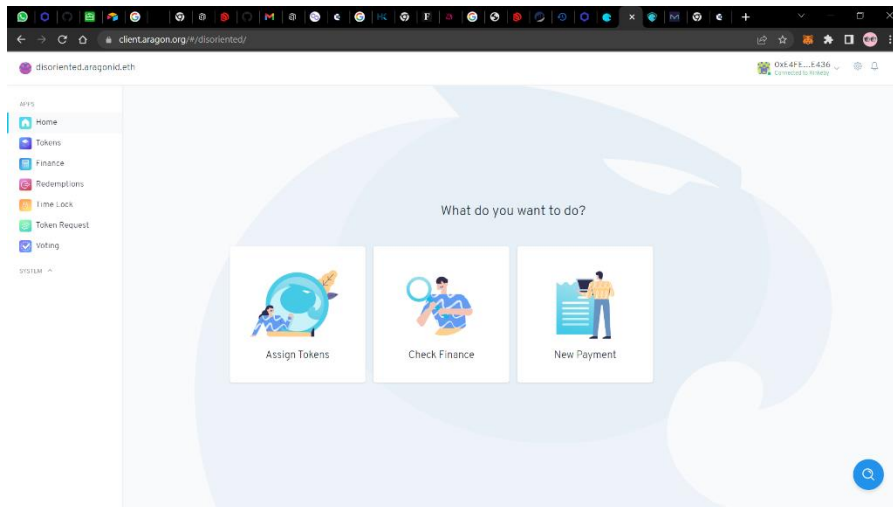
### 3. Provide a name for your organization.

The screenshot shows the 'Configure template' page in the client.aragon.org/#create interface. On the left, a progress indicator shows 50% completion (3/5 steps). The steps are: 1. Dandelion (checked), 2. disoriented.aragonid.eth (checked), 3. Configure template (active), 4. Review information, and 5. Launch organization. The main content area is titled 'Configure template' with the subtitle 'Choose your Voting app settings below.' It features four sliders: 'SUPPORT %' set to 50%, 'MINIMUM APPROVAL %' set to 15%, 'VOTE DURATION' set to 1 Day, 0 Hours, 0 Minutes, and 'VOTE SUFFICE' set to 0 Days, 1 Hour, 0 Minutes. Below these is a 'VOTE DELAY' section set to 1 Day, 0 Hours, 0 Minutes. A note states: 'The support and minimum approval thresholds are strict requirements, such that votes will only pass if they achieve approval percentages greater than these thresholds.' At the bottom are 'Back' and 'Next: Configure template' buttons.

### 4. Configure the template and review the information and then launch the organization.

The screenshot shows the 'All done! Your organization is ready' page in the client.aragon.org/#create interface. On the left, a progress indicator shows 100% completion (5/5 steps). The steps are: 1. Dandelion (checked), 2. disoriented.aragonid.eth (checked), 3. Configure template (checked), 4. Review information (checked), and 5. Launch organization (active). The main content area features an illustration of three people around a globe and the text 'All done! Your organization is ready'. A 'Get started' button is located below the text. At the bottom right is a search icon.

5. We can look into the operations allowed within the organization.



## CREATING A SMART CONTRACT.

- Using Remix, a simple smart contract can be created.
- The smart contract is coded in solidity. A simple program to print “hello world” would like as follows.
- CODE

```
1  pragma solidity ^0.5.0;
2
3  contract ArithValue {
4      uint number;
5      constructor () public {
6          number = 100;
7      }
8      function setNumber(uint value) public {
9          number = value;
10     }
11     function getNumber() public view returns (uint) {
12         return number;
13     }
14     function incrementNumber() public {
15         number = number + 1;
16     }
17     function decrementNumber() public {
18         number = number - 1;
19     }
20 }
```

- OUTPUT

JavaScript VM (London)

VM

ACCOUNT

0x5B3...eddC4 (99.999999%)

GAS LIMIT

3000000

VALUE

0

Wei

CONTRACT

ArithValue - simple.sol

Deploy

☐ Publish to IPFS

OR

At Address

Load contract from Address

Transactions recorded 19

Deployed Contracts

ARITHVALUE AT 0X0FC...9A836 (MEM

decrementNu...

incrementNum...

setNumber

7

getNumber

0: uint256: 8

## - TRANSACTIONS

```
✓ [vm] from: 0x5B3...eddC4 to: ArithValue.incrementNumber() 0x0FC...9A836 value: 0 wei data: 0x273...ea3e3 logs: 0 hash: 0x14e...a93b7
call to ArithValue.getNumber

CALL [call] from: 0x5B38Da6a701c568545dCfc803Fc8875f56beddC4 to: ArithValue.getNumber() data: 0xf2c...9ecd8
transact to ArithValue.setNumber pending ...

✓ [vm] from: 0x5B3...eddC4 to: ArithValue.setNumber(uint256) 0x0FC...9A836 value: 0 wei data: 0x3fb...00007 logs: 0 hash: 0x78e...4de35
call to ArithValue.getNumber

CALL [call] from: 0x5B38Da6a701c568545dCfc803Fc8875f56beddC4 to: ArithValue.getNumber() data: 0xf2c...9ecd8
transact to ArithValue.incrementNumber pending ...

✓ [vm] from: 0x5B3...eddC4 to: ArithValue.incrementNumber() 0x0FC...9A836 value: 0 wei data: 0x273...ea3e3 logs: 0 hash: 0xddd...55001
call to ArithValue.getNumber

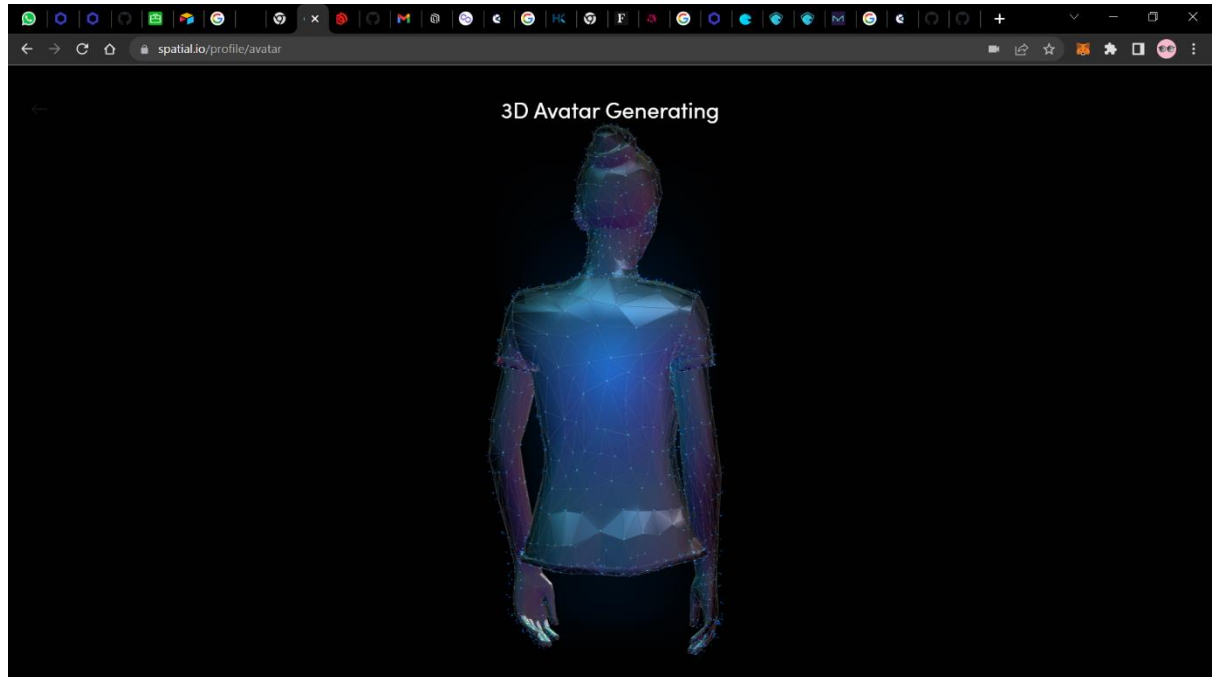
CALL [call] from: 0x5B38Da6a701c568545dCfc803Fc8875f56beddC4 to: ArithValue.getNumber() data: 0xf2c...9ecd8
```

## SPATIAL

### - Entering the metaverse



- Creating your avatar



- About details

