National College of Ireland

Configuration Manual

MSc Research Project
Programme Name

Jake Murphy
Student ID: x16442804

School of Computing
National College of Ireland

Supervisor: Imran Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Jake Murphy……

Student ID: X16642804……..……

Programme: MSc in CyberSecurity – Full Time…… **Year:** 1……..

Module: MSc Research Project……..……

Supervisor: Imran Khan……..……

Submission Due Date: 16/08/2021……..……

Project Title: Blockchain and IOT Security – Can an Ethereum Blockchain be the solution for securing IOT devices.

Word Count: 789…… **Page Count** 10……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Jake Murphy……

Date: 16/08/2021……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□ yes
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□ yes
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□ yes

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

1 Project Dependencies

1. NODE
2. GOLANG
3. GETH
4. Visual studio Code recommended
5. Chrome – Meta Mask extension

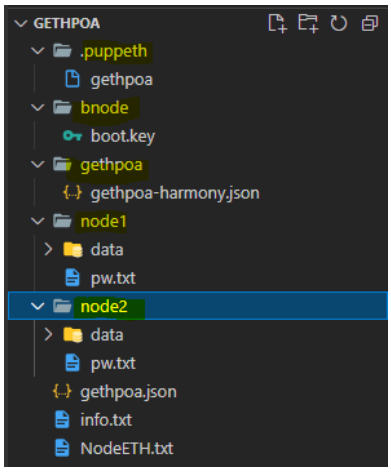
2 Starting ETH BC

Before running any ETH BC locally must ensure that GOLANG and GETH are installed prior. The latest versions should be fine, also for GETH install 'GETH & Tools' to get all of ETHS development tools locally.

GO: https://golang.org/doc/install

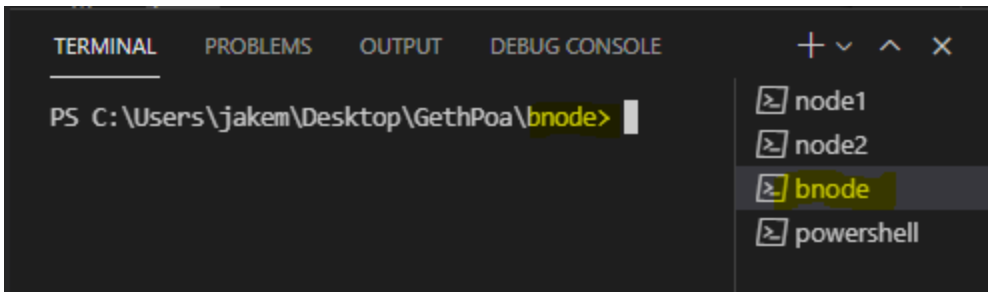
GETH: https://geth.ethereum.org/downloads/

After these have been install successfully can now navigate to the GethPoA directory within a code editor terminal – recommend using Visual studio code can open up multiple PowerShell terminals on the fly .



GethPoA fill strucutre

2.1 Bootnode

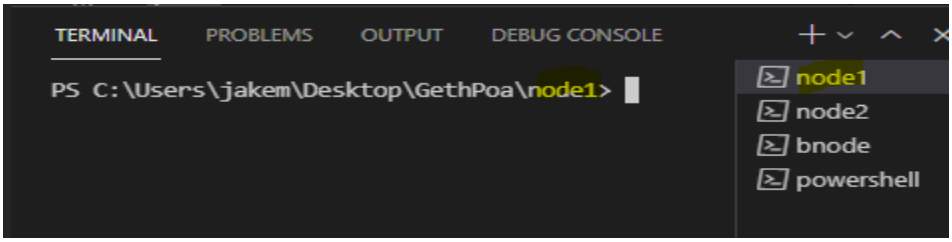


Terminal command: bootnode -nodekey "./boot.key" -verbosity 7 -addr "127.0.0.1:30301"

Result:

```
PS C:\Users\jakem\Desktop\GethPoa\bnode> bootnode -nodekey "./boot.key" -verbosity 7 -addr "127.0.0.1:30301"
enode://4c52e5ab6b8a82e4908b16f2aecdb11ab6028382ba5f0bdb9313c502507dae673bf86f124d133aaabc60276e2648b2b6da59dcc63a7a733c9c82d689e76c392b@127.0.0.1:0?discport=30301
Note: you're using cmd/bootnode, a developer tool.
We recommend using a regular node as bootstrap node for production deployments.
INFO [08-15|23:07:31.194] New local node record                    seq=1 id=9af1322aa6ecb1d4 ip=<nil> udp=0 tcp=0
TRACE[08-15|23:07:57.296] << FINDNODE/v4                           id=188fe0ff851e94d1 addr=127.0.0.1:30303 err="unknown node"
TRACE[08-15|23:07:57.297] << FINDNODE/v4                           id=188fe0ff851e94d1 addr=127.0.0.1:30303 err="unknown node"
TRACE[08-15|23:07:57.807] << FINDNODE/v4                           id=188fe0ff851e94d1 addr=127.0.0.1:30303 err="unknown node"
TRACE[08-15|23:07:57.807] << FINDNODE/v4                           id=188fe0ff851e94d1 addr=127.0.0.1:30303 err="unknown node"
```

2.2 Node 1

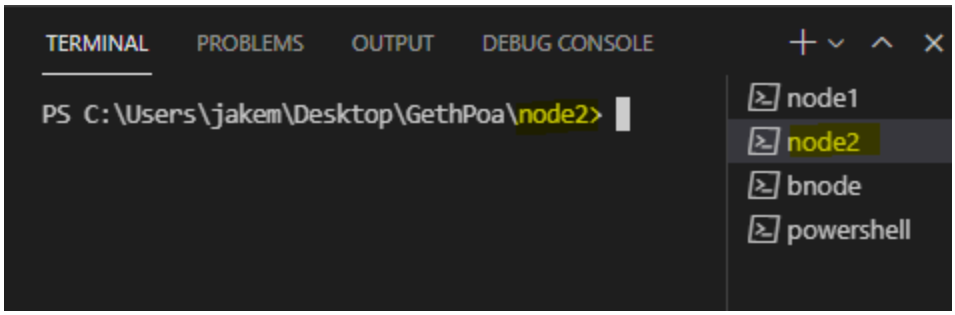


Terminal command: geth --identity "Node1 - Miner" --networkid 14333 --datadir "./data" --bootnodes enode://4c52e5ab6b8a82e4908b16f2aecdb11ab6028382ba5f0bdb9313c502507dae673bf86f124d133aaabc60276e2648b2b6da59dcc63a7a733c9c82d689e76c392b@127.0.0.1:0?discport=30301 --port 30303 --ipcdisable --syncmode full --http --allow-insecure-unlock --http.corsdomain "*" --http.port 8545 --unlock 0x2Ca6b5E088B114ce79d0Bb362A5905C47344e30d --password pw.txt -mine console

Result:

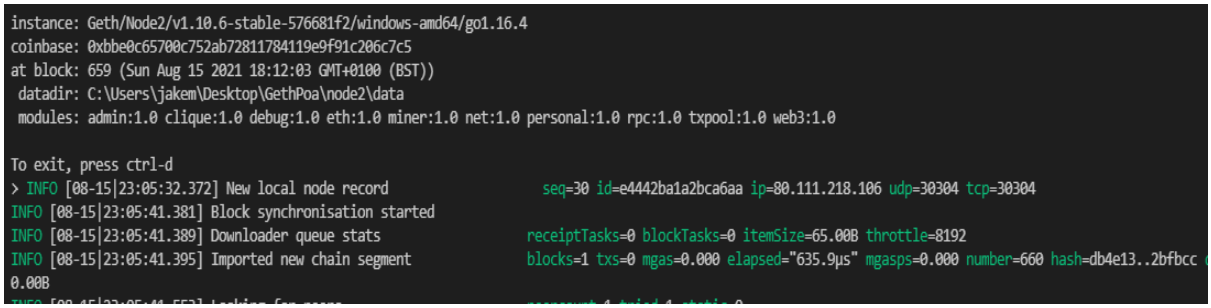
```
Welcome to the Geth JavaScript console!

instance: Geth/Node1 - Miner/v1.10.6-stable-576681f2/windows-amd64/go1.16.4
coinbase: 0x2ca6b5e088b114ce79d0bb362a5905c47344e30d
at block: 662 (Sun Aug 15 2021 23:07:58 GMT+0100 (BST))
 datadir: C:\Users\jakem\Desktop\GethPoa\node1\data
 modules: admin:1.0 clique:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d
> INFO [08-15|23:08:07.510] Looking for peers                        peercount=0 tried=0 static=0
INFO [08-15|23:08:08.007] Successfully sealed new block            number=663 sealhash=23e15e..19795c hash=212a1a..69b8db elapsed=9.956s
INFO [08-15|23:08:08.013] 🔨 mined potential block                  number=663 hash=212a1a..69b8db
INFO [08-15|23:08:08.007] Commit new mining work                   number=664 sealhash=eac0de..f7da01 uncles=0 txs=0 gas=0 fees=0 elapsed=0s
INFO [08-15|23:08:09.114] HTTP server stopped                      endpoint=127.0.0.1:8545
INFO [08-15|23:08:09.118] Ethereum protocol stopped
```

2.3 Node 2



Terminal command: geth --identity "Node2" --networkid 14333 --datadir "./data" --bootnodes enode://4c52e5ab6b8a82e4908b16f2aecdb11ab6028382ba5f0bdb9313c502507dae673bf86f124d133aaabc60276e2648b2b6da59dcc63a7a733c9c82d689e76c392b@127.0.0.1:0?discport=30301 --port 30304 --ipcdisable --syncmode full --http --allow-insecure-unlock --http.corsdomain "*" --http.port 8546 --unlock 0xBBE0c65700C752ab72811784119E9F91C206C7C5 --password pw.txt console

Result:

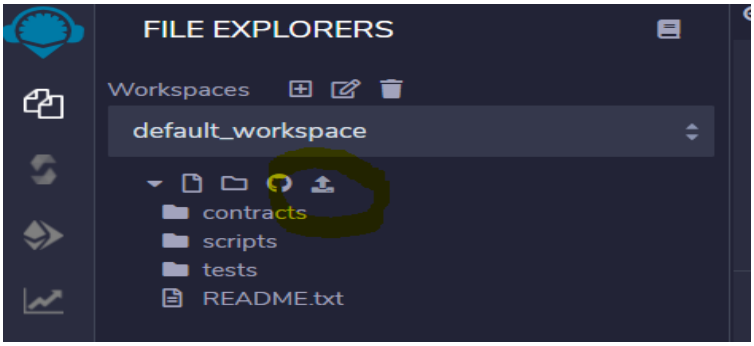


The blockchain should be started in the order of 1) bootnode 2) node1 3) node2. This is to avoid any errors when the BC syncs together.

After running these three commands in there proper directories in separate terminals should be a private ETH BC running locally on your machine.

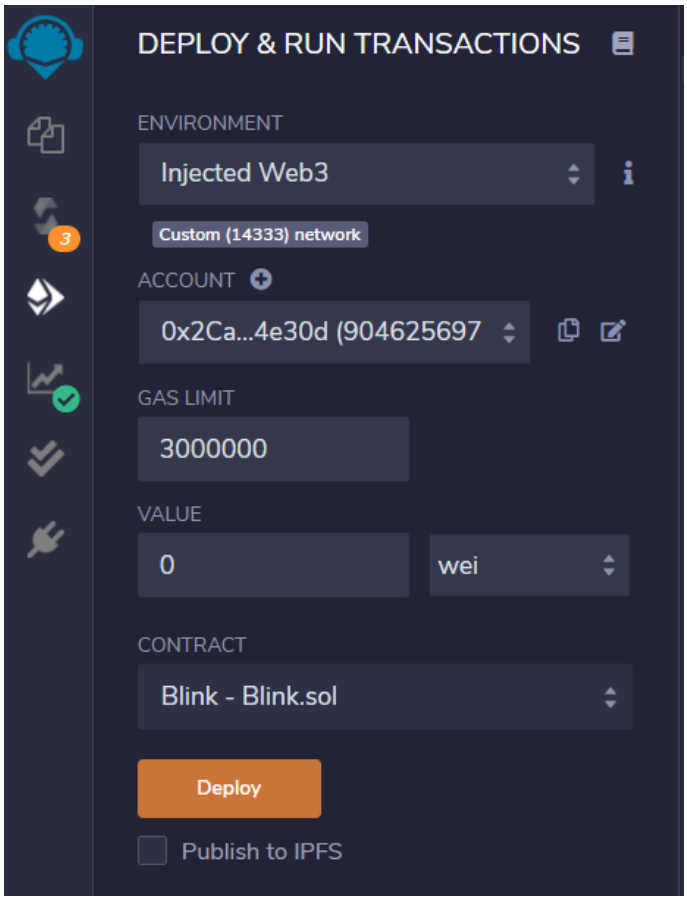
3 REMIX

Remix ETH editor was used to create and test SC's. also allowed developers to connect localhost and work on your smart contracts in a code editor but still run the SC in real time in the browser. But for simplicity upload the Blink.sol file from the DAPP/sol/ directory

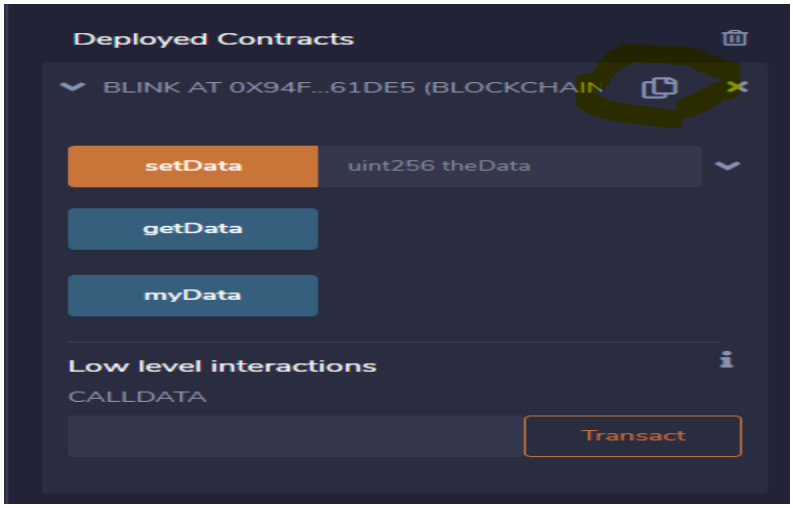


After uploading to SC to Remix can now set up to deploy the contract. For the enviroment delect injected web3, any account as no ether is needed for the transaction (PoA), gas limit

leave at 3000, select the Blink contract. When you click deploy the smart contract will be sent into the BC.



After the contract is in the blockchain need to copy that contract address from the deployed contract and implent into the node application. But first can see the getters and setters of the smart contract and to trigger an event in the BC send data to BC in the 'setData' e.g. 1234.

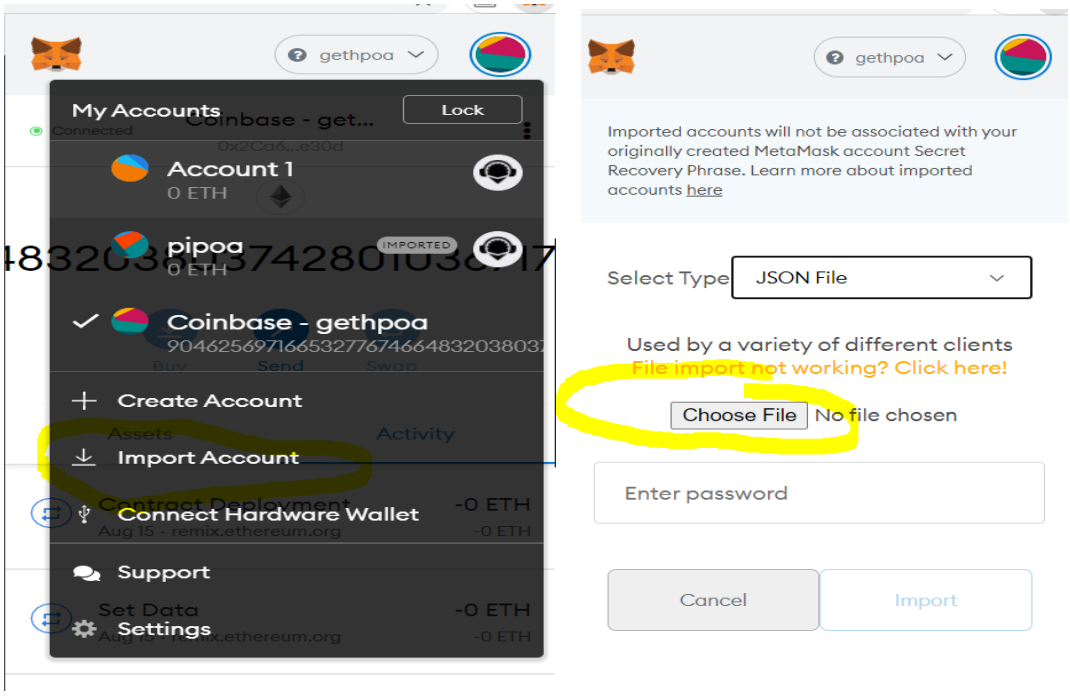


Make sure in the pi/index.js file that the contract adress is the same as the one in the code else application will not accept the request from the SC.

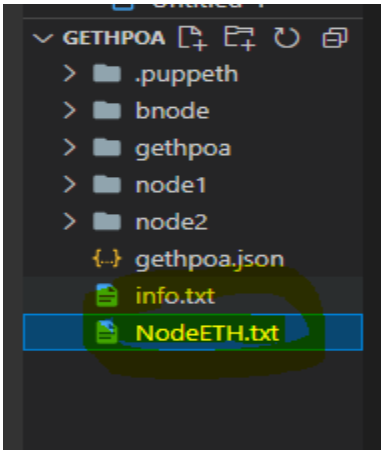
```
// The Smart contract address - Application will only auth this smart contract
var contractAddress = '0xb06578535a1691e6C43952343Cc1CFc4B622250a';
```

4 Meta Mask

After meta mask is installed into the browser can import GethPoA network into meta mask via JSON. The password is the private key of node1 – miner, which is 'node1'



The JSON file can be found in the GethPoA folder as 'NodeETH.txt'



After the JSON file is uploaded and regonises the GethPoA network. It will connect atomatically when the BC is running on the machine.

5 Running node app

Navigate to pi directory and in the terminal of that directory type:

- Npm install

This will install all the project dependencies via node.

After all the dependencies are installed can run the application by staying within the pi directory and type:

- Npm start

This will start the node application and wait for request from the blockchain via SC.

OPTIONAL

1. Raspbian OS
2. Install node v7 binary files onto a raspberry pi (raspberry pi 3/4).
3. Load the node app onto the pi
4. Npm install
5. Npm run

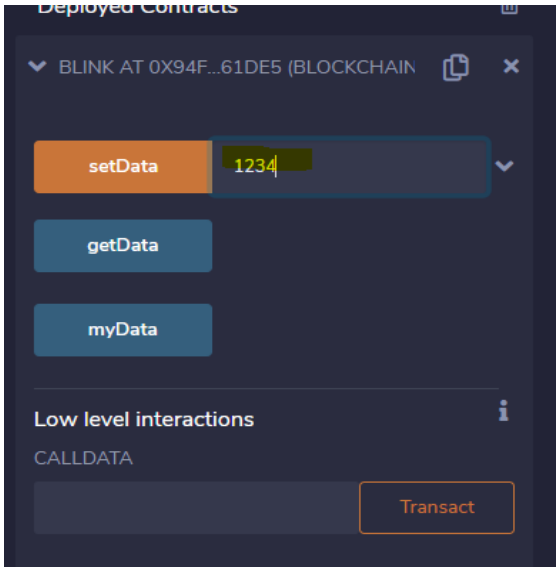
The app should runs on a pi as well but I ran into some of the web3 libraries not installing properly (Node is experimental on the Pi)

If running the Pi will have to change the web3 address in the pi/index.js to the IP of that machine on port 8545 instead of localhost.

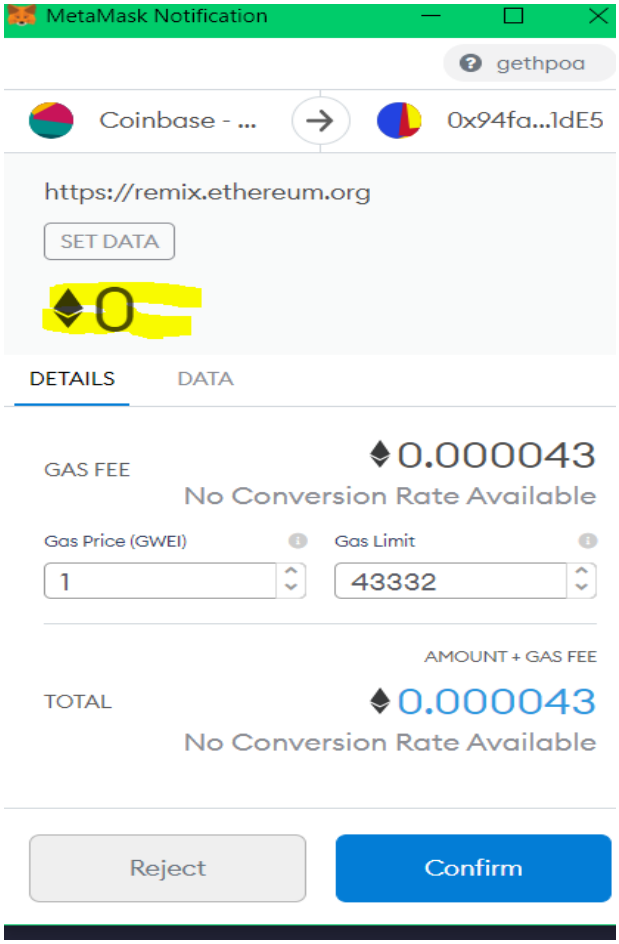
```
web3.setProvider(new web3.providers.HttpProvider("http://127.0.0.1:8545")); // GethPoA netowrk is being ran from localhost:8545
```

6 Final Step

Now that the BC is running, meta mask is setup the GethPoa, Remix has the SC and the node app is installed and updated on the machine just need to make a meta mask transaction to carry out a secure transaction to the pi via SC. After the SC is deployed into the network and meta mask is connected to the local BC can carry out a transaction.



After sending some data through ‘setData’ will be prompt by the gethpoa wallet and asked to carry out a transaction. Zero funds are needed as it’s a PoA BC and doesn’t rely on ether.



After you confirm the transation through meta mask will be proccessed by the GethPoA BC then send the request to the node applcaition where a secure reuqest is made to the raspberry pi applciation.

```
[nodemon] starting `babel-node index.js`
 - Blink Function Initiated -
PI-Node on port 3030
[nodemon] restarting due to changes...
[nodemon] starting `babel-node index.js`
 - Blink Function Initiated -
PI-Node on port 3030
{
  address: '0xb06578535a1691e6c43952343cc1cfc4b622250a',
  blockNumber: 655,
  transactionHash: '0x7fc0b2de1c767139db291fce8a8b1bfbf804911c884d4a25689697882813fe79',
  transactionIndex: 0,
  blockHash: '0x32c7739088704a0eec0bbd981666dc65573b9d401a589e9e1af623d16516c6b4',
  logIndex: 0,
  removed: false,
  event: 'blinkEvent',
  args: { data: BigNumber { s: 1, e: 3, c: [Array] } }
}
Turning on LEDS - On
Turning off LEDs - Sleep
Turning on LEDS - On
Turning off LEDs - Sleep
Turning on LEDS - On
Turning off LEDs - Sleep
Turning on LEDS - On
Turning off LEDs - Sleep
```