Configuration Manual

MSc Research Project
Cybersecurity

Mariusz Graczyk
Student ID: x20197446

School of Computing
National College of Ireland

Supervisor: Mr. Ross Spelman

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Mariusz Graczyk
Student ID:	x20197446
Programme:	MSCCYBETOP Year: 2024
Module:	Research Project
Supervisor:	Mr. Ross Spelman
Submission Due Date:	12 August 2024
Project Title:	Enhancing SDN Access Control with Private Ethereum Blockchain
Word Count:	510 Page Count 8

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: Mariusz Graczyk

Date: 16 September 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
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.	☐

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

Mariusz Graczyk
Student ID: x20197446

1 Main Dependencies Used

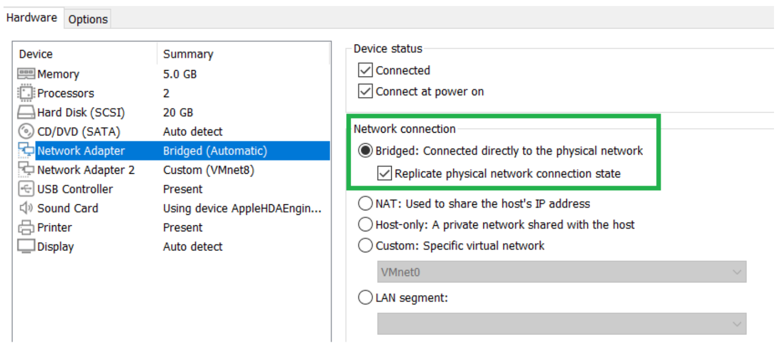
a. VMware Workstation 16 Pro
b. Ryu package version 4.34
c. Ubuntu release 18.04
d. Python3 version 3.7.5
e. Python version 2.7.17 – for deploying Mininet scripts
f. Mininet version 2.2.2
g. Remix IDE v. 0.52.0
h. Geth and Puppeth version: 1.9.24-stable
i. Go Version: 1.15.5
j. Chrome or Firefox with Metamask extension plugin
k. Raspberry Pi 4 with installed Open vSwitch v.2.17.1
l. USB to Ethernet adapter for the Raspberry Pi devices
m. Visual Studio Code 1.92.1

2 Other Dependencies Used

a. VirtualBox v. 6.1
b. Kali Linux release 2024.2
c. Ganache v. 2.7.1
d. Notepad++ v8.4.8 (with Compare plugin)
e. Matplotlib library for Python

3 Deployment of Ubuntu VM

a. Install Ubuntu in VMware. The network adapter should be in bridged mode



Device	Summary
Memory	5.0 GB
Processors	2
Hard Disk (SCSI)	20 GB
CD/DVD (SATA)	Auto detect
Network Adapter	Bridged (Automatic)
Network Adapter 2	Custom (VMnet8)
USB Controller	Present
Sound Card	Using device AppleHDAEngin...
Printer	Present
Display	Auto detect

Device status
✓ Connected
✓ Connect at power on

Network connection
● Bridged: Connected directly to the physical network
✓ Replicate physical network connection state
○ NAT: Used to share the host's IP address
○ Host-only: A private network shared with the host
○ Custom: Specific virtual network
VMnet0
○ LAN segment:

Figure 1: Virtual network adapter in Bridged mode

4 Deployment of Ryu and Mininet

b. Install Ryu package and Mininet emulator in the Ubuntu VM:

```
sudo pip3 install ryu
sudo apt install mininet
```

c. Install Open vSwitch on Raspberry Pi 4[1]

d. Run Ryu Manager with a simple L3 switch [1]:

```
ryu-manager ryu.app.simple_switch_13
```

The script for `simple_switch_13` was borrowed from C²S Consulting[2]

e. Launch Mininet hosts to confirm connectivity to the controller [1]:

```
sudo mn --controller remote,ip=127.0.0.1  --switch
ovsk,protocols=OpenFlow13 --mac --ipbase=10.0.0.0/24 --topo single,4
```

f. Launch the final target Mininet topology from the Python script:

```
sudo python ~/mininet/custom/Nat_Mininet_topology.py
```

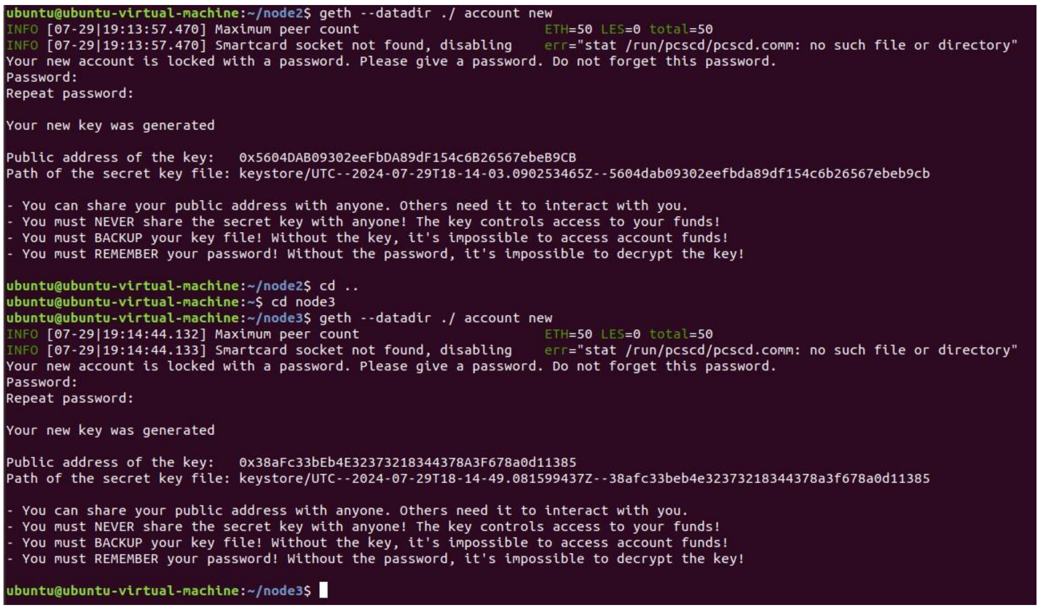
The script `Nat_Mininet_topology.py` was adapted from [3] and [4].

g. Ryu-Manager command that launches the controller alongside the *SDNEther* app, with the required certificates for TLS encryption.

```
ryu-manager --observe-links ~/ryu/flowmanager/flowmanager.py
ryu.app.L3_simple_switch_13 sdnether.py --ctl-privkey
/var/lib/openvswitch/pki/controllerca/ctl-privkey.pem --ctl-cert
/var/lib/openvswitch/pki/controllerca/ctl-cert.pem --ca-certs
/var/lib/openvswitch/pki/switchca/cacert.pem
```

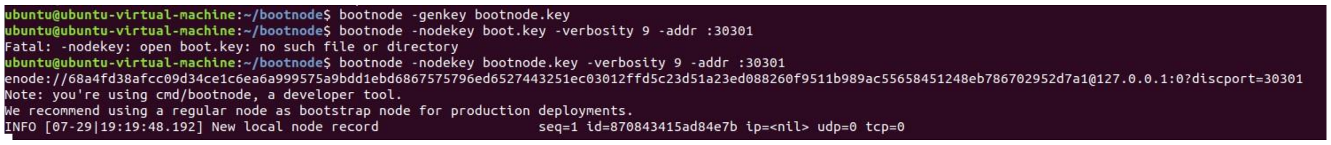
[1] https://shantoroy.com/openflow/how-to-configure-raspberry-pi-as-open-flow-switch/
[2] https://www.obriain.com/training/sdn/RYU_example_scripts_v2.0.1.tgz

5 Deployment of PoA Ethereum Network



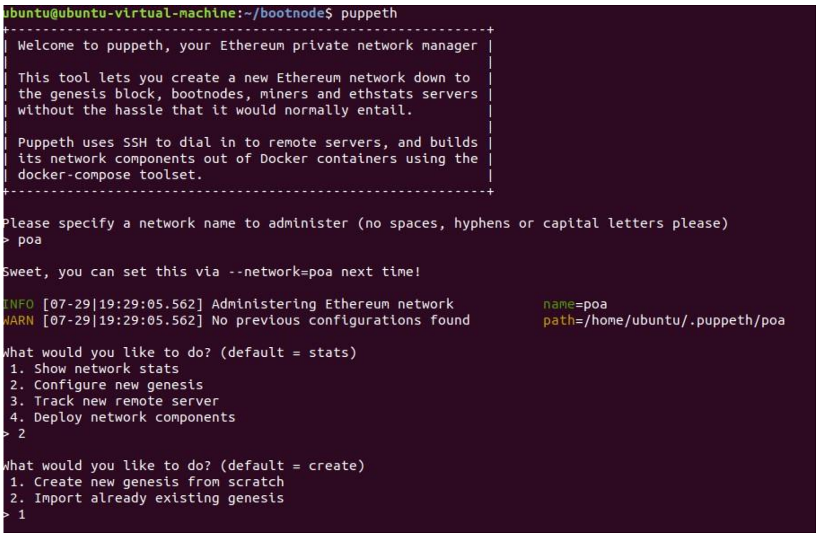
```
ubuntu@ubuntu-virtual-machine:~/node2$ geth --datadir ./ account new
INFO [07-29|19:13:57.470] Maximum peer count                       ETH=50 LES=0 total=50
INFO [07-29|19:13:57.470] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
Your new account is locked with a password. Please give a password. Do not forget this password.
Password:
Repeat password:

Your new key was generated

Public address of the key:   0x5604DAB09302eeFbDA89dF154c6B26567ebeB9CB
Path of the secret key file: keystore/UTC--2024-07-29T18-14-03.090253465Z--5604dab09302eefbda89df154c6b26567ebeb9cb

- You can share your public address with anyone. Others need it to interact with you.
- You must NEVER share the secret key with anyone! The key controls access to your funds!
- You must BACKUP your key file! Without the key, it's impossible to access account funds!
- You must REMEMBER your password! Without the password, it's impossible to decrypt the key!

ubuntu@ubuntu-virtual-machine:~/node2$ cd ..
ubuntu@ubuntu-virtual-machine:~$ cd node3
ubuntu@ubuntu-virtual-machine:~/node3$ geth --datadir ./ account new
INFO [07-29|19:14:44.132] Maximum peer count                       ETH=50 LES=0 total=50
INFO [07-29|19:14:44.133] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
Your new account is locked with a password. Please give a password. Do not forget this password.
Password:
Repeat password:

Your new key was generated

Public address of the key:   0x38aFc33bEb4E32373218344378A3F678a0d11385
Path of the secret key file: keystore/UTC--2024-07-29T18-14-49.081599437Z--38afc33beb4e32373218344378a3f678a0d11385

- You can share your public address with anyone. Others need it to interact with you.
- You must NEVER share the secret key with anyone! The key controls access to your funds!
- You must BACKUP your key file! Without the key, it's impossible to access account funds!
- You must REMEMBER your password! Without the password, it's impossible to decrypt the key!

ubuntu@ubuntu-virtual-machine:~/node3$ █
```

Figure 2: Initial activation of a new account for each of the three nodes



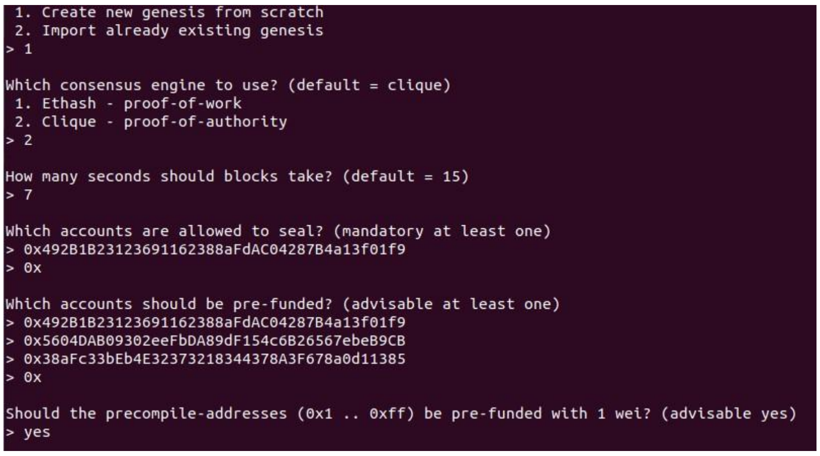
```
ubuntu@ubuntu-virtual-machine:~/bootnode$ bootnode -genkey bootnode.key
ubuntu@ubuntu-virtual-machine:~/bootnode$ bootnode -nodekey boot.key -verbosity 9 -addr :30301
Fatal: -nodekey: open boot.key: no such file or directory
ubuntu@ubuntu-virtual-machine:~/bootnode$ bootnode -nodekey bootnode.key -verbosity 9 -addr :30301
enode://68a4fd38afcc09d34ce1c6ea6a999575a9bdd1ebd6867575796ed6527443251ec03012ffd5c23d51a23ed088260f9511b989ac55658451248eb786702952d7a1@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 [07-29|19:19:48.192] New local node record                    seq=1 id=870843415ad84e7b ip=<nil> udp=0 tcp=0
```

Figure 3: New key for the bootnode is generated and the bootnode is launched



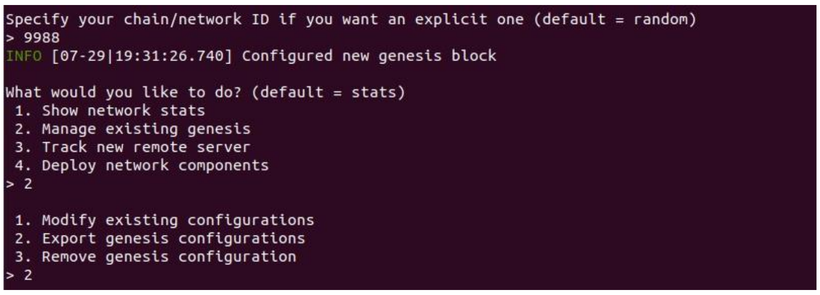
```
buntu@ubuntu-virtual-machine:~/bootnode$ puppeth
+-----------------------------------------------------------+
| Welcome to puppeth, your Ethereum private network manager |
|                                                           |
| This tool lets you create a new Ethereum network down to  |
| the genesis block, bootnodes, miners and ethstats servers |
| without the hassle that it would normally entail.         |
|                                                           |
| Puppeth uses SSH to dial in to remote servers, and builds |
| its network components out of Docker containers using the |
| docker-compose toolset.                                   |
+-----------------------------------------------------------+

lease specify a network name to administer (no spaces, hyphens or capital letters please)
 poa

weet, you can set this via --network=poa next time!

NFO [07-29|19:29:05.562] Administering Ethereum network           name=poa
ARN [07-29|19:29:05.562] No previous configurations found         path=/home/ubuntu/.puppeth/poa

hat would you like to do? (default = stats)
 1. Show network stats
 2. Configure new genesis
 3. Track new remote server
 4. Deploy network components
 2

hat would you like to do? (default = create)
 1. Create new genesis from scratch
 2. Import already existing genesis
 1
```

Figure 4: New Genesis file is created



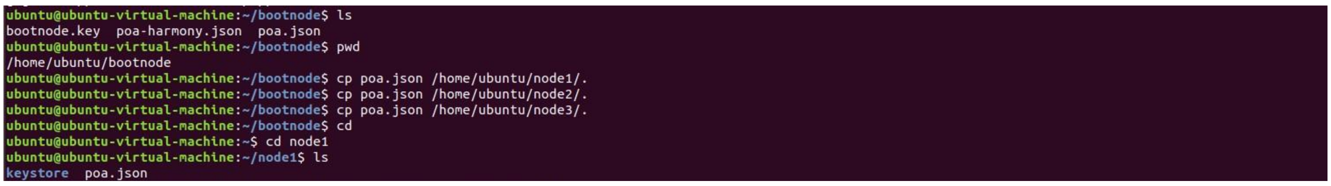
```
 1. Create new genesis from scratch
 2. Import already existing genesis
> 1

Which consensus engine to use? (default = clique)
 1. Ethash - proof-of-work
 2. Clique - proof-of-authority
> 2

How many seconds should blocks take? (default = 15)
> 7

Which accounts are allowed to seal? (mandatory at least one)
> 0x492B1B23123691162388aFdAC04287B4a13f01f9
> 0x

Which accounts should be pre-funded? (advisable at least one)
> 0x492B1B23123691162388aFdAC04287B4a13f01f9
> 0x5604DAB09302eeFbDA89dF154c6B26567ebeB9CB
> 0x38aFc33bEb4E32373218344378A3F678a0d11385
> 0x

Should the precompile-addresses (0x1 .. 0xff) be pre-funded with 1 wei? (advisable yes)
> yes
```

Figure 5: PoA Clique engine is selected and the miner node is designated. All three accounts are pre-funded



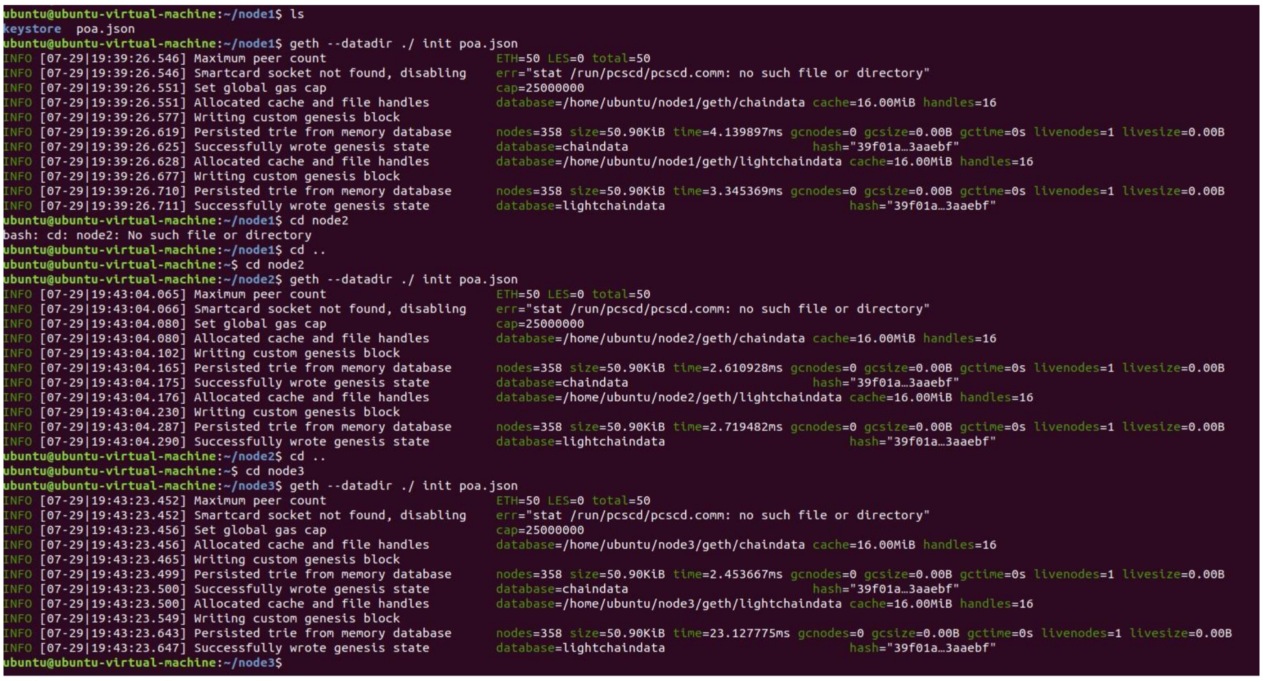
```
Specify your chain/network ID if you want an explicit one (default = random)
> 9988
INFO [07-29|19:31:26.740] Configured new genesis block

What would you like to do? (default = stats)
 1. Show network stats
 2. Manage existing genesis
 3. Track new remote server
 4. Deploy network components
> 2

 1. Modify existing configurations
 2. Export genesis configurations
 3. Remove genesis configuration
> 2
```

Figure 6: The network ID is created and the Genesis configuration is exported



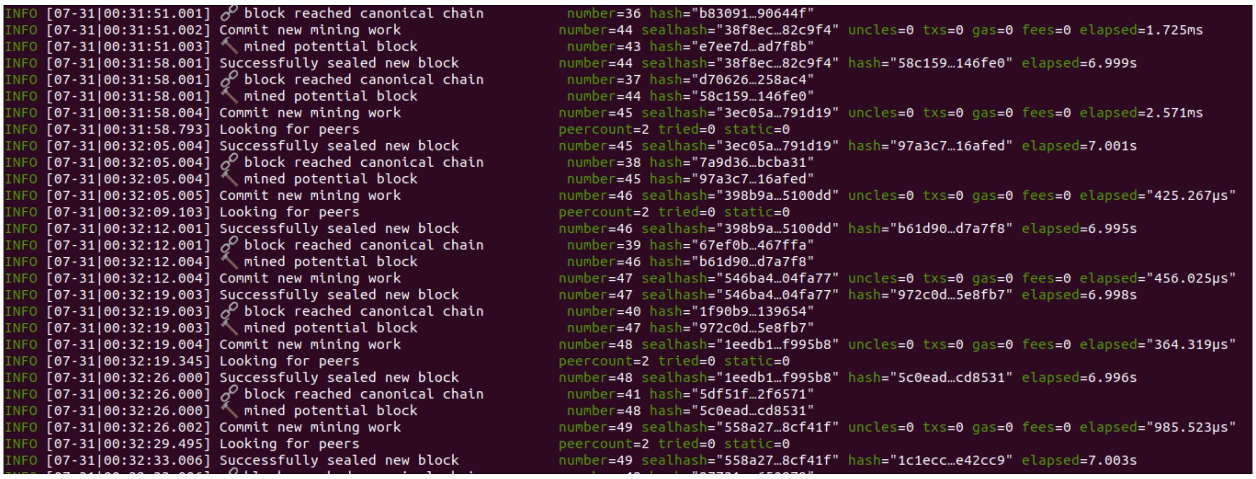
```
ubuntu@ubuntu-virtual-machine:~/bootnode$ ls
bootnode.key  poa-harmony.json  poa.json
ubuntu@ubuntu-virtual-machine:~/bootnode$ pwd
/home/ubuntu/bootnode
ubuntu@ubuntu-virtual-machine:~/bootnode$ cp poa.json /home/ubuntu/node1/.
ubuntu@ubuntu-virtual-machine:~/bootnode$ cp poa.json /home/ubuntu/node2/.
ubuntu@ubuntu-virtual-machine:~/bootnode$ cp poa.json /home/ubuntu/node3/.
ubuntu@ubuntu-virtual-machine:~/bootnode$ cd
ubuntu@ubuntu-virtual-machine:~$ cd node1
ubuntu@ubuntu-virtual-machine:~/node1$ ls
keystore  poa.json
```

Figure 7: The Genesis file is copied to each node's folder and the Genesis is initialised for each node.



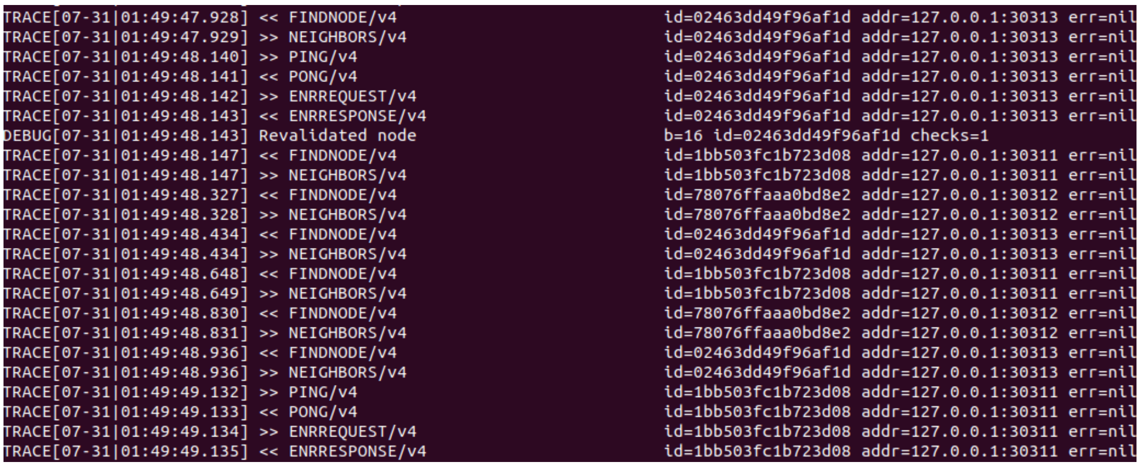
```
ubuntu@ubuntu-virtual-machine:~/node1$ ls
keystore  poa.json
ubuntu@ubuntu-virtual-machine:~/node1$ geth --datadir ./ init poa.json
INFO [07-29|19:39:26.546] Maximum peer count                       ETH=50 LES=0 total=50
INFO [07-29|19:39:26.546] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
INFO [07-29|19:39:26.551] Set global gas cap                       cap=25000000
INFO [07-29|19:39:26.551] Allocated cache and file handles         database=/home/ubuntu/node1/geth/chaindata cache=16.00MiB handles=16
INFO [07-29|19:39:26.577] Writing custom genesis block
INFO [07-29|19:39:26.619] Persisted trie from memory database      nodes=358 size=50.90KiB time=4.139897ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [07-29|19:39:26.625] Successfully wrote genesis state         database=chaindata                         hash="39f01a…3aaebf"
INFO [07-29|19:39:26.628] Allocated cache and file handles         database=/home/ubuntu/node1/geth/lightchaindata cache=16.00MiB handles=16
INFO [07-29|19:39:26.677] Writing custom genesis block
INFO [07-29|19:39:26.710] Persisted trie from memory database      nodes=358 size=50.90KiB time=3.345369ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [07-29|19:39:26.711] Successfully wrote genesis state         database=lightchaindata                         hash="39f01a…3aaebf"
ubuntu@ubuntu-virtual-machine:~/node1$ cd node2
bash: cd: node2: No such file or directory
ubuntu@ubuntu-virtual-machine:~/node1$ cd ..
ubuntu@ubuntu-virtual-machine:~$ cd node2
ubuntu@ubuntu-virtual-machine:~/node2$ geth --datadir ./ init poa.json
INFO [07-29|19:43:04.065] Maximum peer count                       ETH=50 LES=0 total=50
INFO [07-29|19:43:04.066] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
INFO [07-29|19:43:04.080] Set global gas cap                       cap=25000000
INFO [07-29|19:43:04.080] Allocated cache and file handles         database=/home/ubuntu/node2/geth/chaindata cache=16.00MiB handles=16
INFO [07-29|19:43:04.102] Writing custom genesis block
INFO [07-29|19:43:04.165] Persisted trie from memory database      nodes=358 size=50.90KiB time=2.610928ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [07-29|19:43:04.175] Successfully wrote genesis state         database=chaindata                         hash="39f01a…3aaebf"
INFO [07-29|19:43:04.176] Allocated cache and file handles         database=/home/ubuntu/node2/geth/lightchaindata cache=16.00MiB handles=16
INFO [07-29|19:43:04.230] Writing custom genesis block
INFO [07-29|19:43:04.287] Persisted trie from memory database      nodes=358 size=50.90KiB time=2.719482ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [07-29|19:43:04.290] Successfully wrote genesis state         database=lightchaindata                         hash="39f01a…3aaebf"
ubuntu@ubuntu-virtual-machine:~/node2$ cd ..
ubuntu@ubuntu-virtual-machine:~$ cd node3
ubuntu@ubuntu-virtual-machine:~/node3$ geth --datadir ./ init poa.json
INFO [07-29|19:43:23.452] Maximum peer count                       ETH=50 LES=0 total=50
INFO [07-29|19:43:23.452] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
INFO [07-29|19:43:23.456] Set global gas cap                       cap=25000000
INFO [07-29|19:43:23.456] Allocated cache and file handles         database=/home/ubuntu/node3/geth/chaindata cache=16.00MiB handles=16
INFO [07-29|19:43:23.465] Writing custom genesis block
INFO [07-29|19:43:23.499] Persisted trie from memory database      nodes=358 size=50.90KiB time=2.453667ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [07-29|19:43:23.500] Successfully wrote genesis state         database=chaindata                         hash="39f01a…3aaebf"
INFO [07-29|19:43:23.500] Allocated cache and file handles         database=/home/ubuntu/node3/geth/lightchaindata cache=16.00MiB handles=16
INFO [07-29|19:43:23.549] Writing custom genesis block
INFO [07-29|19:43:23.643] Persisted trie from memory database      nodes=358 size=50.90KiB time=23.127775ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [07-29|19:43:23.647] Successfully wrote genesis state         database=lightchaindata                         hash="39f01a…3aaebf"
ubuntu@ubuntu-virtual-machine:~/node3$
```

Figure 8: Each node is initialised with the Genesis file



```
INFO [07-31|00:31:51.001] 🔗 block reached canonical chain          number=36 hash="b83091…90644f"
INFO [07-31|00:31:51.002] Commit new mining work                   number=44 sealhash="38f8ec…82c9f4" uncles=0 txs=0 gas=0 fees=0 elapsed=1.725ms
INFO [07-31|00:31:51.003] 🔨 mined potential block                  number=43 hash="e7ee7d…ad7f8b"
INFO [07-31|00:31:58.001] Successfully sealed new block            number=44 sealhash="38f8ec…82c9f4" hash="58c159…146fe0" elapsed=6.999s
INFO [07-31|00:31:58.001] 🔗 block reached canonical chain          number=37 hash="d70626…258ac4"
INFO [07-31|00:31:58.001] 🔨 mined potential block                  number=44 hash="58c159…146fe0"
INFO [07-31|00:31:58.004] Commit new mining work                   number=45 sealhash="3ec05a…791d19" uncles=0 txs=0 gas=0 fees=0 elapsed=2.571ms
INFO [07-31|00:31:58.793] Looking for peers                        peercount=2 tried=0 static=0
INFO [07-31|00:32:05.004] Successfully sealed new block            number=45 sealhash="3ec05a…791d19" hash="97a3c7…16afed" elapsed=7.001s
INFO [07-31|00:32:05.004] 🔗 block reached canonical chain          number=38 hash="7a9d36…bcba31"
INFO [07-31|00:32:05.004] 🔨 mined potential block                  number=45 hash="97a3c7…16afed"
INFO [07-31|00:32:05.005] Commit new mining work                   number=46 sealhash="398b9a…5100dd" uncles=0 txs=0 gas=0 fees=0 elapsed="425.267µs"
INFO [07-31|00:32:09.103] Looking for peers                        peercount=2 tried=0 static=0
INFO [07-31|00:32:12.001] Successfully sealed new block            number=46 sealhash="398b9a…5100dd" hash="b61d90…d7a7f8" elapsed=6.995s
INFO [07-31|00:32:12.001] 🔗 block reached canonical chain          number=39 hash="67ef0b…467ffa"
INFO [07-31|00:32:12.004] 🔨 mined potential block                  number=46 hash="b61d90…d7a7f8"
INFO [07-31|00:32:12.004] Commit new mining work                   number=47 sealhash="546ba4…04fa77" uncles=0 txs=0 gas=0 fees=0 elapsed="456.025µs"
INFO [07-31|00:32:19.003] Successfully sealed new block            number=47 sealhash="546ba4…04fa77" hash="972c0d…5e8fb7" elapsed=6.998s
INFO [07-31|00:32:19.003] 🔗 block reached canonical chain          number=40 hash="1f90b9…139654"
INFO [07-31|00:32:19.003] 🔨 mined potential block                  number=47 hash="972c0d…5e8fb7"
INFO [07-31|00:32:19.004] Commit new mining work                   number=48 sealhash="1eedb1…f995b8" uncles=0 txs=0 gas=0 fees=0 elapsed="364.319µs"
INFO [07-31|00:32:19.345] Looking for peers                        peercount=2 tried=0 static=0
INFO [07-31|00:32:26.000] Successfully sealed new block            number=48 sealhash="1eedb1…f995b8" hash="5c0ead…cd8531" elapsed=6.996s
INFO [07-31|00:32:26.000] 🔗 block reached canonical chain          number=41 hash="5df51f…2f6571"
INFO [07-31|00:32:26.000] 🔨 mined potential block                  number=48 hash="5c0ead…cd8531"
INFO [07-31|00:32:26.002] Commit new mining work                   number=49 sealhash="558a27…8cf41f" uncles=0 txs=0 gas=0 fees=0 elapsed="985.523µs"
INFO [07-31|00:32:29.495] Looking for peers                        peercount=2 tried=0 static=0
INFO [07-31|00:32:33.006] Successfully sealed new block            number=49 sealhash="558a27…8cf41f" hash="1c1ecc…e42cc9" elapsed=7.003s
```

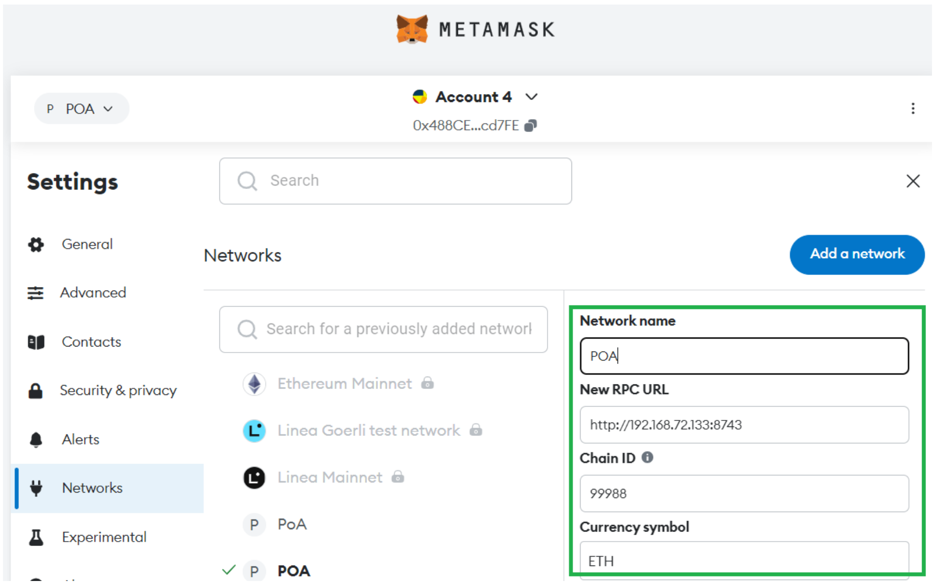
Figure 9: Miner (sealer) node is launched



```
TRACE[07-31|01:49:47.928] << FINDNODE/v4                            id=02463dd49f96af1d addr=127.0.0.1:30313 err=nil
TRACE[07-31|01:49:47.929] >> NEIGHBORS/v4                           id=02463dd49f96af1d addr=127.0.0.1:30313 err=nil
TRACE[07-31|01:49:48.140] >> PING/v4                                id=02463dd49f96af1d addr=127.0.0.1:30313 err=nil
TRACE[07-31|01:49:48.141] << PONG/v4                                id=02463dd49f96af1d addr=127.0.0.1:30313 err=nil
TRACE[07-31|01:49:48.142] >> ENRREQUEST/v4                          id=02463dd49f96af1d addr=127.0.0.1:30313 err=nil
TRACE[07-31|01:49:48.143] << ENRRESPONSE/v4                         id=02463dd49f96af1d addr=127.0.0.1:30313 err=nil
DEBUG[07-31|01:49:48.143] Revalidated node                          b=16 id=02463dd49f96af1d checks=1
TRACE[07-31|01:49:48.147] << FINDNODE/v4                            id=1bb503fc1b723d08 addr=127.0.0.1:30311 err=nil
TRACE[07-31|01:49:48.147] >> NEIGHBORS/v4                           id=1bb503fc1b723d08 addr=127.0.0.1:30311 err=nil
TRACE[07-31|01:49:48.327] << FINDNODE/v4                            id=78076ffaaa0bd8e2 addr=127.0.0.1:30312 err=nil
TRACE[07-31|01:49:48.328] >> NEIGHBORS/v4                           id=78076ffaaa0bd8e2 addr=127.0.0.1:30312 err=nil
TRACE[07-31|01:49:48.434] << FINDNODE/v4                            id=02463dd49f96af1d addr=127.0.0.1:30313 err=nil
TRACE[07-31|01:49:48.434] >> NEIGHBORS/v4                           id=02463dd49f96af1d addr=127.0.0.1:30313 err=nil
TRACE[07-31|01:49:48.648] << FINDNODE/v4                            id=1bb503fc1b723d08 addr=127.0.0.1:30311 err=nil
TRACE[07-31|01:49:48.649] >> NEIGHBORS/v4                           id=1bb503fc1b723d08 addr=127.0.0.1:30311 err=nil
TRACE[07-31|01:49:48.830] << FINDNODE/v4                            id=78076ffaaa0bd8e2 addr=127.0.0.1:30312 err=nil
TRACE[07-31|01:49:48.831] >> NEIGHBORS/v4                           id=78076ffaaa0bd8e2 addr=127.0.0.1:30312 err=nil
TRACE[07-31|01:49:48.936] << FINDNODE/v4                            id=02463dd49f96af1d addr=127.0.0.1:30313 err=nil
TRACE[07-31|01:49:48.936] >> NEIGHBORS/v4                           id=02463dd49f96af1d addr=127.0.0.1:30313 err=nil
TRACE[07-31|01:49:49.132] >> PING/v4                                id=1bb503fc1b723d08 addr=127.0.0.1:30311 err=nil
TRACE[07-31|01:49:49.133] << PONG/v4                                id=1bb503fc1b723d08 addr=127.0.0.1:30311 err=nil
TRACE[07-31|01:49:49.134] >> ENRREQUEST/v4                          id=1bb503fc1b723d08 addr=127.0.0.1:30311 err=nil
TRACE[07-31|01:49:49.135] << ENRRESPONSE/v4                         id=1bb503fc1b723d08 addr=127.0.0.1:30311 err=nil
```

Figure 10: Bootnode is launched and it facilitates communication among all the nodes

6 Deploying the Smart Contract with Remix and Metamask



Network name

POA

New RPC URL

http://192.168.72.133:8743

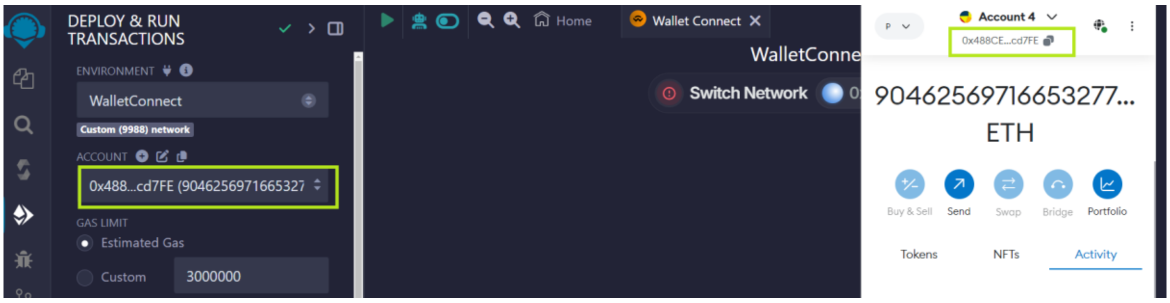
Chain ID

99988

Currency symbol

ETH

Figure 11: PoA network needs to be set up with URL, network ID and currency


DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: WalletConnect

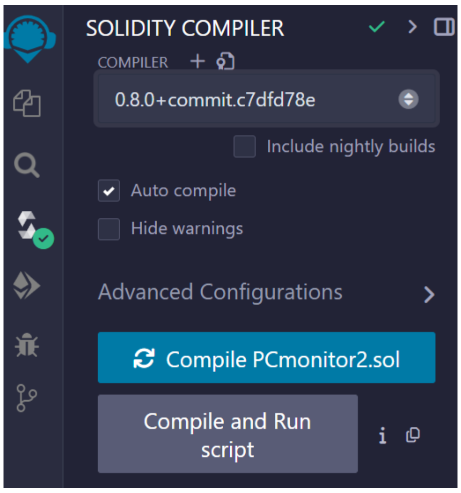
Custom (9988) network

ACCOUNT: 0x488...cd7FE (9046256971665327

GAS LIMIT: Estimated Gas

Custom 3000000

Figure 12: Metamask account needs to be connected with Remix via WalletConnect


SOLIDITY COMPILER

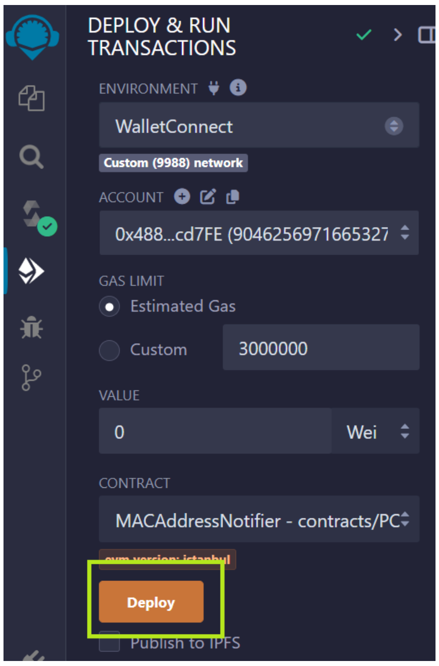
COMPILER: 0.8.0+commit.c7dfd78e

☐ Include nightly builds

☑ Auto compile

☐ Hide warnings

Figure 13: Smart Contract code needs to be compiled with the correct version of compiler corresponding to the correct pragma solidity ^0.8.0;


DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: WalletConnect

Custom (9988) network

ACCOUNT: 0x488...cd7FE (9046256971665327

GAS LIMIT: Estimated Gas

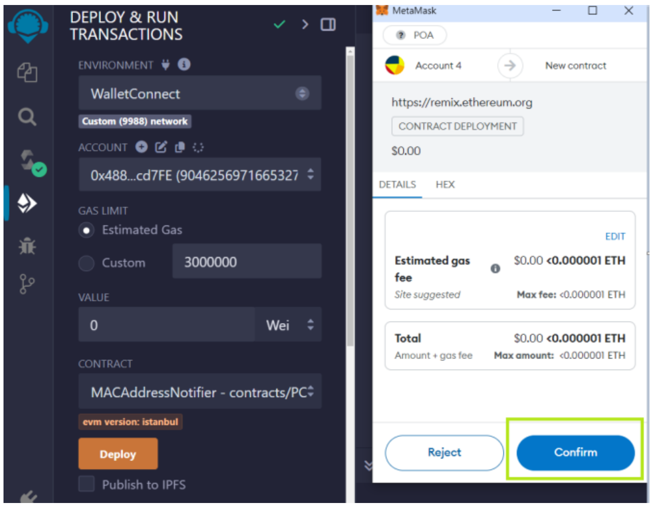
Custom 3000000

VALUE: 0 Wei

CONTRACT: MACAddressNotifier - contracts/PC

evm version: istanbul

Figure 14: Smart Contract is deployed to the BC


DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: WalletConnect

Custom (9988) network

ACCOUNT: 0x488...cd7FE (9046256971665327

GAS LIMIT: Estimated Gas

Custom: 3000000

VALUE: 0 Wei

CONTRACT: MACAddressNotifier - contracts/PC

evm version: istanbul

Deploy

Publish to IPFS

MetaMask

POA

Account 4 → New contract

https://remix.ethereum.org

CONTRACT DEPLOYMENT

$0.00

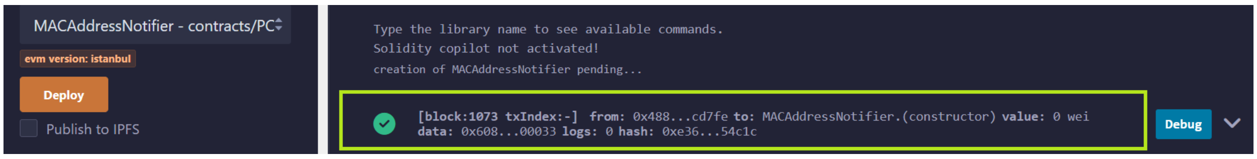
DETAILS HEX

EDIT

Estimated gas fee	$0.00 <0.000001 ETH
Site suggested	Max fee: <0.000001 ETH
Total	$0.00 <0.000001 ETH
Amount + gas fee	Max amount: <0.000001 ETH

Reject Confirm

Figure 15: Final stage: contract deployment confirmation in Metamask


MACAddressNotifier - contracts/PC

evm version: istanbul

Deploy

Publish to IPFS

```
Type the library name to see available commands.
Solidity copilot not activated!
creation of MACAddressNotifier pending...

[block:1073 txIndex:-] from: 0x488...cd7fe to: MACAddressNotifier.(constructor) value: 0 wei
data: 0x608...00033 logs: 0 hash: 0xe36...54c1c
```

Debug

Figure 16: Successful transaction has been confirmed

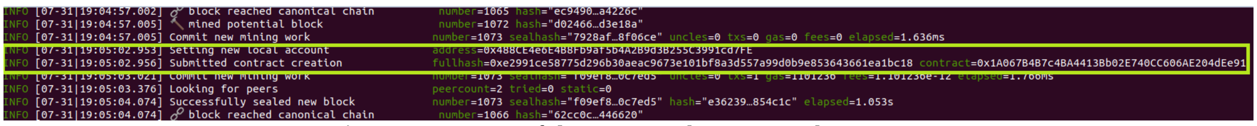
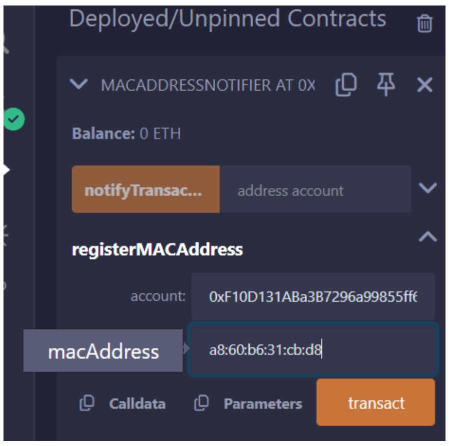

```
INFO [07-31|19:04:57.002] 🔗 block reached canonical chain         number=1065 hash="ec9490…a4226c"
INFO [07-31|19:04:57.005] 🔨 mined potential block                 number=1072 hash="d02466…d3e18a"
INFO [07-31|19:04:57.005] Commit new mining work                  number=1073 sealhash="7928af…8f06ce" uncles=0 txs=0 gas=0 fees=0 elapsed=1.636ms
INFO [07-31|19:05:02.953] Setting new local account               address=0x488CE4e6E4B8Fb9af5b4A2B9d3B255C3991cd7FE
INFO [07-31|19:05:02.956] Submitted contract creation             fullhash=0xe2991ce58775d296b30aeac9673e101bf8a3d557a99d0b9e853643661ea1bc18 contract=0x1A067B4B7c4BA4413Bb02E740CC606AE204dEe91
INFO [07-31|19:05:03.021] Commit new mining work                  number=1073 sealhash="f09ef8…0c7ed5" uncles=0 txs=1 gas=1101236 fees=1.101236e-12 elapsed=1.766ms
INFO [07-31|19:05:03.376] Looking for peers                       peercount=2 tried=0 static=0
INFO [07-31|19:05:04.074] Successfully sealed new block           number=1073 sealhash="f09ef8…0c7ed5" hash="e36239…854c1c" elapsed=1.053s
INFO [07-31|19:05:04.074] 🔗 block reached canonical chain         number=1066 hash="62cc0c…446620"
```

Figure 17: Successful contract submision in the BC


Deployed/Unpinned Contracts

MACADDRESSNOTIFIER AT 0X

Balance: 0 ETH

notifyTransac... address account

registerMACAddress

account: 0xF10D131ABa3B7296a99855ff€

macAddress a8:60:b6:31:cb:d8

Calldata Parameters transact

Figure 18: End-user MAC address and its correspoding BC account address can be registered in the SC from Remix IDE

References

[1] “RYU SDN Testbed Manual Diarmuid Ó Briain engcore advancing technology.” Available: https://www.obriain.com/training/sdn/RYU_Soft_Testbed_v2.0.pdf. (Accessed 12 Jun., 2024).

[2] “Setup TLS Connection — Ryu 4.34 documentation,” Readthedocs.io, 2022. https://ryu.readthedocs.io/en/latest/tls.html (Accessed 30 Jun., 2024).

[3] “Open vSwitch with SSL and Mininet,” Tech and Trains, Apr. 27, 2014. https://techandtrains.com/2014/04/27/open-vswitch-with-ssl-and-mininet/ (accessed Aug. 9, 2024).

[4] Mininet, “mininet/examples/natnet.py at master mininet/mininet,” GitHub, 2024. https://github.com/mininet/mininet/blob/master/examples/natnet.py (accessed Jun. 2, 2024).