Title	Harmony Blockchain
Description	Description of the ebook.
Date	May 13, 2022
Author	Arashtad
Author URI	https://arashtad.com

The main purpose of all blockchains have been decentralizing the peer to peer transactions. After Bitcoin, Ethereum managed to propose a more useful blockchain for decentralized applications. As the blockchain users grew in number, all networks faced the problem of scalability. Since then, all the developers have worked hard to solve this issue. Among all the ecosystems, Harmony has provided one of the best solutions to solve this problem by proposing the Sharding mechanism. This mechanism solves the problem of scalability in addition to the speed of the transactions.

In this article we are going to get familiar with the Harmony blockchain, its architectures, goals, distinctions compared to other blockchains and so on. We are also going to work on some of the scripts related to the transactions and the validaters. Moreover, we will create accounts, wallets and get some test One tokens.

Harmony Architecture

Harmony is one the very few blockchains that has solved the 3 main challenges of all the blockchains. Those challenges are security, scalablility and decentralization. Many cryptocurrencies have tried to solve the trade-off between scalability and decentralization and most have failed. Harmony uses sharding and cross-sharding mechanism to solve the this trade-off. However, Harmony is not the first and not the only blockchain that uses this technique. Zilliqa was the first cryptocurrency to introduce the sharding mechanism with the difference that it uses the Proof Of Work (POW) algorithm to verify the transactions which was not as energy efficient as POS (Proof of Stake).

There are 2 elements that make Harmony secure. One is using VRF (Verifiable Random Function) method which selects validators and nodes in a random and unpredictable way. And the second one is the process called the Effective Proof of Stake which is a sharding based proof of stake that lets validators with more staked tokens run more nodes. Harmony does not have voting and instead of that the validators are selected based on the tokens that they stake.

Harmony uses 4 types of shards: shard 0 which is responsible for transmitting data between different shards, shard 1, 2 and 3. Each one of the shards has a limit of 250 validators. Any one of the Harmony shards are capable of reaching 1000 transactions per second. A blockchain validator is someone who is responsible for verifying transactions on a blockchain. Like Ethereum, Harmony uses solidity and Ethers.js to create Dapps. As a result, Ethereum developers feel no difference developing Dapps on Harmony blockchain.

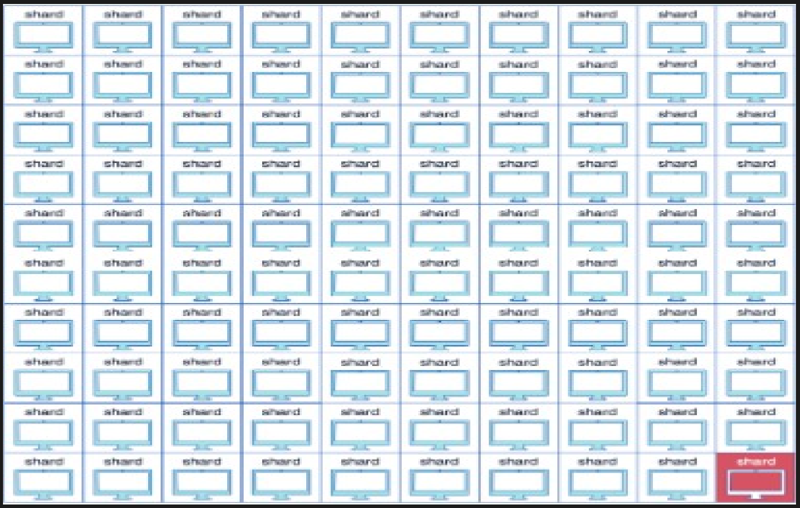
What is special about harmony sharding?

Sharding is a type of database partitioning that separates databases into smaller, faster and more optimized parts called data shards. Each of the shards play the same role as the entire database or network with the difference using shards instead of an entire network provides a higher speed for the transactions, more secure network and a more optimized blockchain.

Harmony uses deep sharding and not only does it use sharding for the software, it also uses it for the network layers. It is quite clear that some other cryptocurrencies use sharding for software layer but it is only the Harmony that uses this technique for the network layer. Shard 0 also known as Beacon shard is responsible for connecting the validators to the shards by using the random generation protocol.

What is 1% attack in a sharded network?

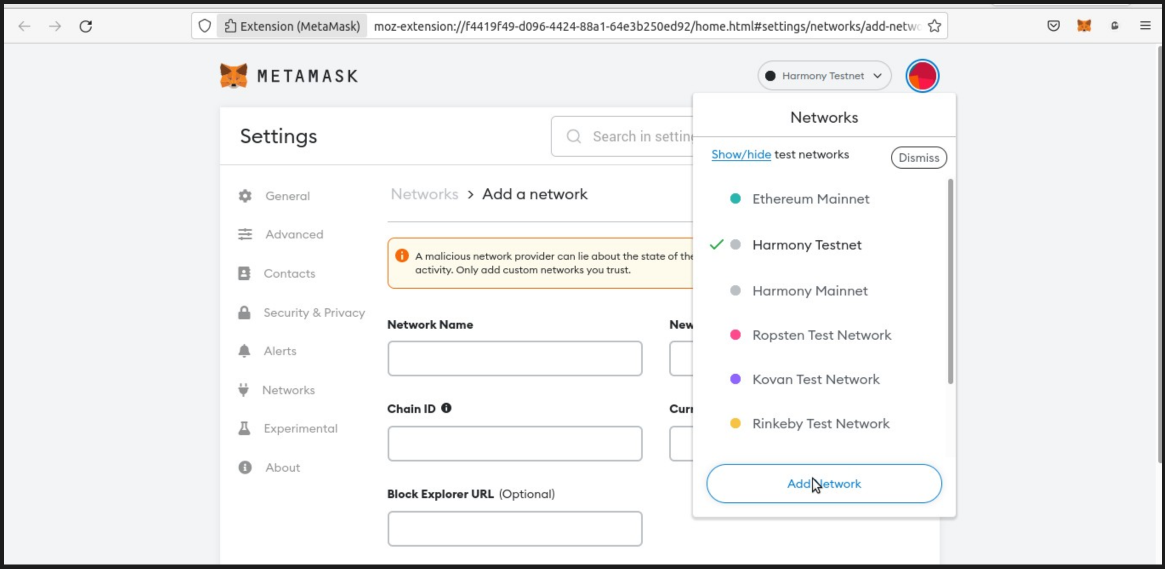
Although a sharded network is much more secure than a non-sharded one, still we cannot say that this kind of network is immune to attack. A 1% shard attack occurs when malicious stakers have more than 1/3 (one third) of the voting shares in shard. On the Harmony network, sharding conditions are changed at intervals or Epochs. This further randomizes the shards handling different transactions and consequently increases the security.



Adding harmony test-net and mainnet to metamask:

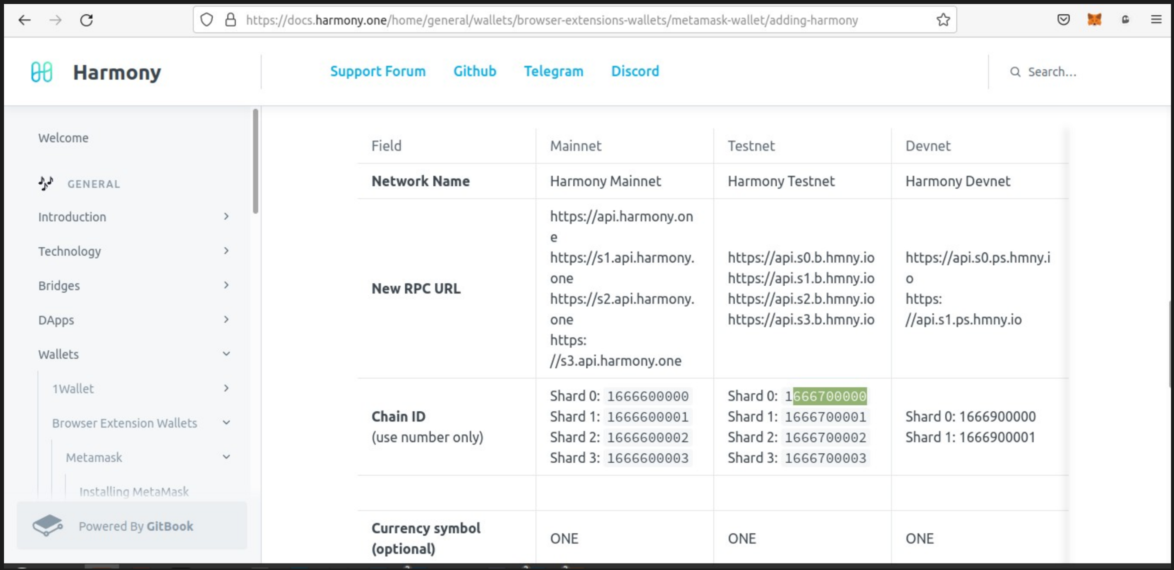
Harmony used to have its own wallet as a chrome extension but according to its official documentation website at (https://docs.harmony.one/home/general/wallets/browser-extensions-wallets/one-wallet) the said wallet is no longer supported. And the Metamask is used instead.

To connect to Metamask, you need to first head over to the Metamask extension in your browser and select the network list on top:



A malicious network provider can lie about the state of the activity. Only add custom networks you trust.

Then enter the specifications of the network that you want and the shard that you prefer in the empty boxes. You can find these specifications such as the network name, RPC URL, Chain ID and the token symbol from the Harmony official documentation website at https://docs.harmony.one/home/general/wallets/browser-extensions-wallets/metamask-wallet/adding-harmony .



Field	Mainnet	Testnet	Devnet
Network Name	Harmony Mainnet	Harmony Testnet	Harmony Devnet
New RPC URL	https://api.harmony.one https://s1.api.harmony.one https://s2.api.harmony.one https://s3.api.harmony.one	https://api.s0.b.hmny.io https://api.s1.b.hmny.io https://api.s2.b.hmny.io https://api.s3.b.hmny.io	https://api.s0.ps.hmny.io https://api.s1.ps.hmny.io
Chain ID (use number only)	Shard 0: 1666600000 Shard 1: 1666600001 Shard 2: 1666600002 Shard 3: 1666600003	Shard 0: 1666700000 Shard 1: 1666700001 Shard 2: 1666700002 Shard 3: 1666700003	Shard 0: 1666900000 Shard 1: 1666900001
Currency symbol (optional)	ONE	ONE	ONE

After you have entered all of the specifications, click save and now you can have your desired network with the shard that you want. The network can either be the test-net or the main-net.

Running Harmony Dapp:

Now , it is time to run the Harmony Dapp on our operating system. Using this Dapp you can stake ONE tokens and become one of the validators of the network. To run this Dapp, you can simply open a VS Code window with a terminal or open a terminal and it enter the following commands:

```
mkdir Harmony_net

cd Harmony_net
```

With the above commands, you can create a directory and then enter the created directory. Now, if you are on Linux, first make sure that you have npm and yarn installed and then enter the following commands separately in the terminal:

```
sudo npm install -g yo

sudo npm install -g generator-harmony-dapp
```

If you are on Mac or Windows, also make sure that you have node and yarn installed and then enter the commands without sudo. In other words, enter the followings in the terminal one after the other has finished:

```
npm install -g yo

npm install -g generator-harmony-dapp
```

To create the Dapp, enter this command in the terminal:

```
yo harmony dapp
```

If you see the following result at the end, you can be sure that eveything has been successful up to now:

```
============================================================
======= Welcome to Harmony Demo ========
============================================================
yarn install v1.22.18
[1/4] Resolving packages...
success Already up-to-date.
Done in 1.70s.
```

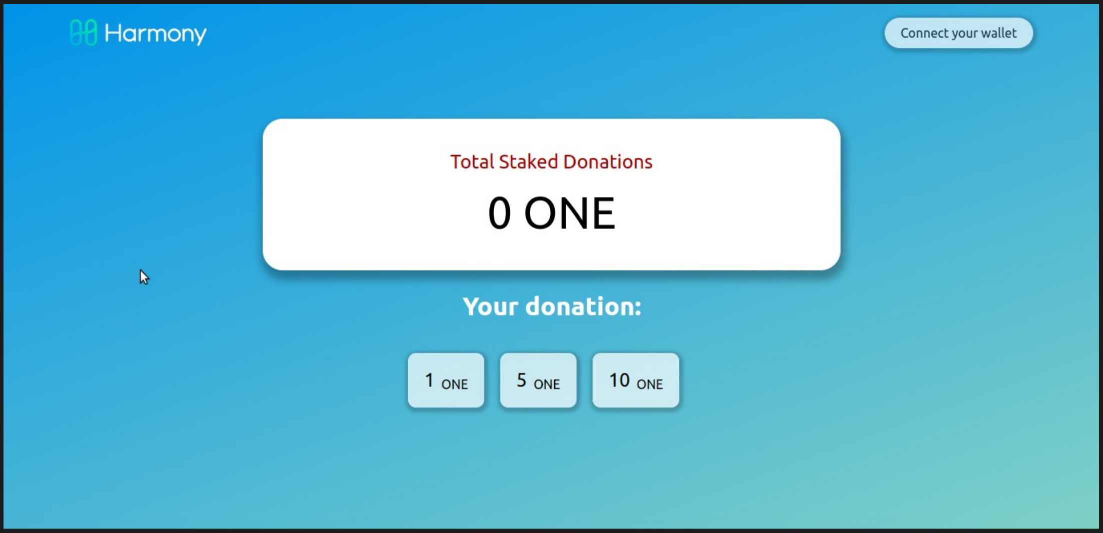
To initialize the Dapp, use the following command:

```
Yarn autoinit
```

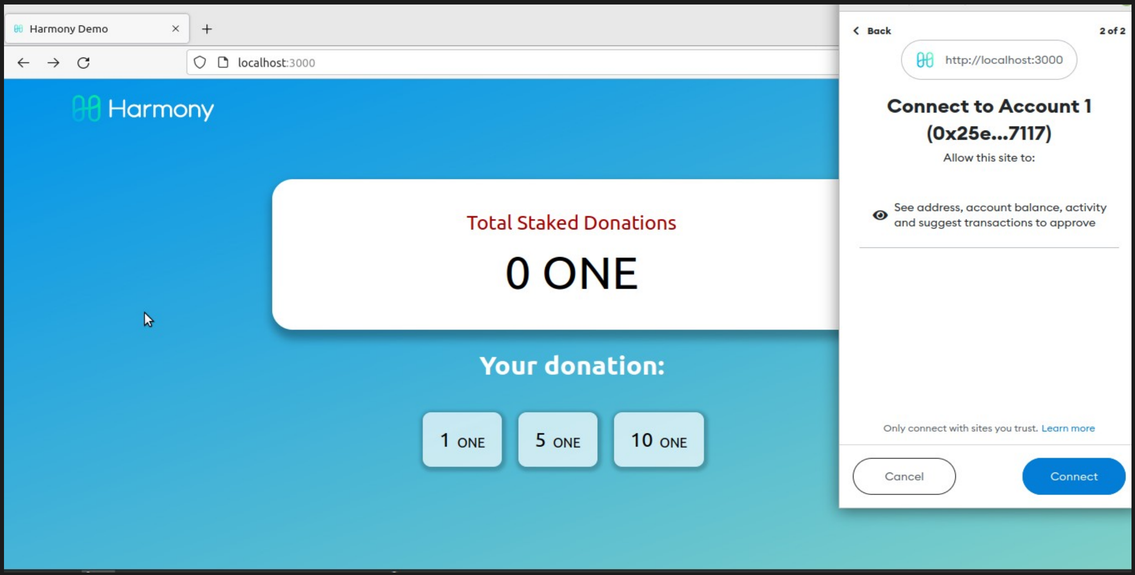
After the process of initialization finishes, you can enter the following in an additional bash or terminal:

```
Yarn start
```

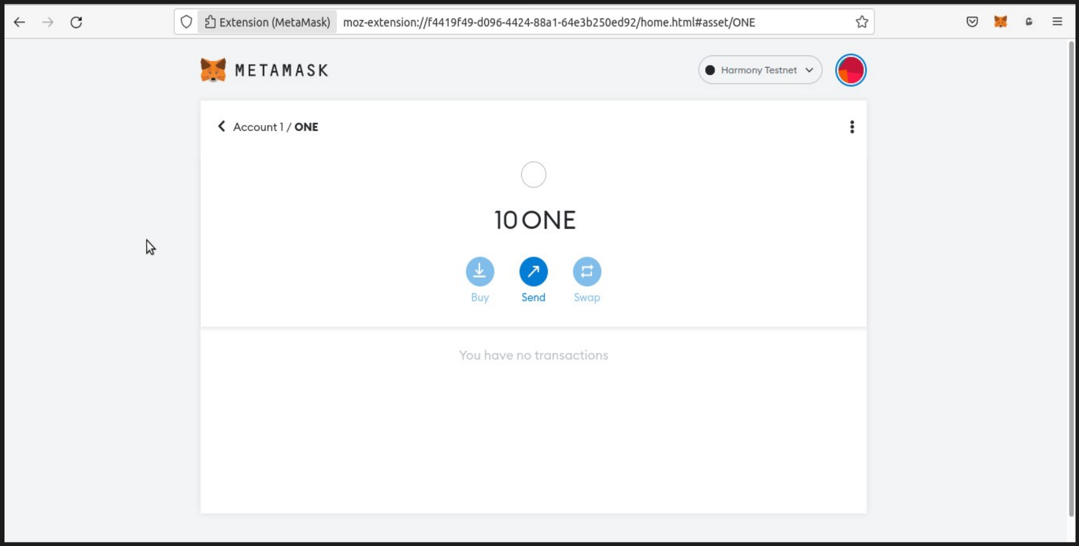
And you will see that after a few seconds a browser pops up and opens the localhost which shows that you have successfully installed the Harmony Dapp:



We should now connect our Metamask wallet to the Dapp. To do so, you need to first click connect to wallet button and select Metamask. Afterwards, you should confirm the connection:


Harmony Demo
localhost:3000
Harmony
Total Staked Donations
0 ONE
Your donation:
1 ONE
5 ONE
10 ONE
Back
2 of 2
http://localhost:3000
Connect to Account 1 (0x25e...7117)
Allow this site to:
See address, account balance, activity and suggest transactions to approve
Only connect with sites you trust. Learn more
Cancel
Connect

Once that is done, you will be able to see the address of your Harmony account on the top right of the page. If you are on test-net, you can also get some test ONE tokens from the Harmony test-net faucet. To do so, let's head over to the faucet at https://faucet.pops.one/ and enter the account address you have got in the blank form, then click on the *Send Me* button. After a few seconds, you will be able to see that you have got 10 ONE tokens in your Metamask Harmony Test-net account.



You can now simply stake some of your test tokens using the Dapp to become a validator of the Harmony test network.

signing a transaction using python:

You can sign the transactions with python using 2 different libraries. One of them is PYHMY and the other is Web3. If you have followed our tutorials on Web3 and Ethereum, you are completely familiar with the way the transactions are created, signed and deployed. It is also worth mentioning that if we use Brownie module the process will be even much easier than that of Web3.

In this tutorial, we are going to sign our transactions with PYHMY. If you have read our article ***Harmony Blockchain: Getting Familiar with It Using Python***, and have followed along with the steps taken in that article, you must now have all the dependencies installed and following scripts are the continuation of the said article. So let's get started with our scripts:

```
from pyhmy import account
from pyhmy import staking
from pyhmy.validator import Validator

test_net = 'https://api.s0.b.hmny.io' # shard 0
test_net_shard_1 = 'https://api.s1.b.hmny.io'

all_validators = staking.get_all_validator_addresses(
                                 endpoint=test_net)
print(len(all_validators))
validator = Validator(all_validators[0])
print(validator)
validator.load_from_blockchain(test_net)

tx_hash = validator.sign_create_validator_transaction(
            nonce = 2,
            gas_price = 1,
            gas_limit = 100,
            private_key =  '4edef2c24995d15b0e25cbd152fb0e2c05
                               d3b79b9c2afd134e6f59f91bf99e48',
            chain_id = None).rawTransaction.hex()

print(tx_hash)
```

In the above scripts, we have connected to the shard0 of the test-net, then we have fetched the list of all the validators and have printed out the number of all of them. Since we need a validator to address to create and sign a transaction, we have used the first one in the list of all the validators on shard 0. Finally using the selected validator, we have created and signed the transaction. The result of the code can be obtained by running the below command in the terminal:

```
Python3 transaction.py
```

Transaction.py is the name of the script file. The result is as follows:

```
6136
{"validator-addr":
"one15uwk6umfxxyljn8ptq27fjjsdd0yl0ar26fd04", "name": null,
"website": null, "security-contact": null, "identity": null,
"details": null, "amount": null, "min-self-delegation": null,
"max-total-delegation": null, "rate": null, "max-rate": null,
"max-change-rate": null, "bls-public-keys": []}
0xf9016280f9011b94a71d6d73693189f94ce15815e4ca506b5e4fbfa3f893
964861726d6f6e7954465f44315f56616c696461746f72ae4861726d6f6e79
54465f44315f56616c696461746f722d323032302d30352d32392d30392d35
342d33332d383437936874747070733a2f2f6861726d6f6e792e6f6e65977661
6c696461746f7240736f6d6577686572652e636f6da056616c696461746f72
2067656e6572617465642062792048617226d6f6e795446ddc9880163457 85d
8a0000c98806f05b59d3b20000c887b1a2bc2ec500008a021e19e0c9bab240
00008a02544faa778090e00000f1b0a07fd0abe605303eb56bc5e6307c57c6
c0f30ae0ea5eda8107137bb92aef4b31780954d7311eec290c9c1623c64c3c
948a021e19e0c9bab240000024a0ada8edc3bfbda8e364b6b7fb843198ba36
20367c21ae7974f4501491b02892aba065c7404c543d15f14076cd5440c1e8
d1252769876d99c230182d301e8e0e2103
```

Conclusion

In this article, we have got familiar with the Harmony Blockchain, what it is, what it does, why it does it and what makes Harmony unique. We have discussed about the trade-offs of all the blockchains and how the sharding mechanism in Harmony has solved these trade-offs. Furthermore, we have learned about the shard attack and how Harmony has secured its network against these attacks.

In addition to the architecture of the Harmony Blockchain, we have learned how to run a Dapp on our browser for staking ONE tokens as well as how we can add Harmony test-net and main-net to our Metamask wallet. Moreover, we got some test ONE token from Harmony test-net faucet. In the end, we worked on a script where we managed to create and sign a transaction using PYHMY library.