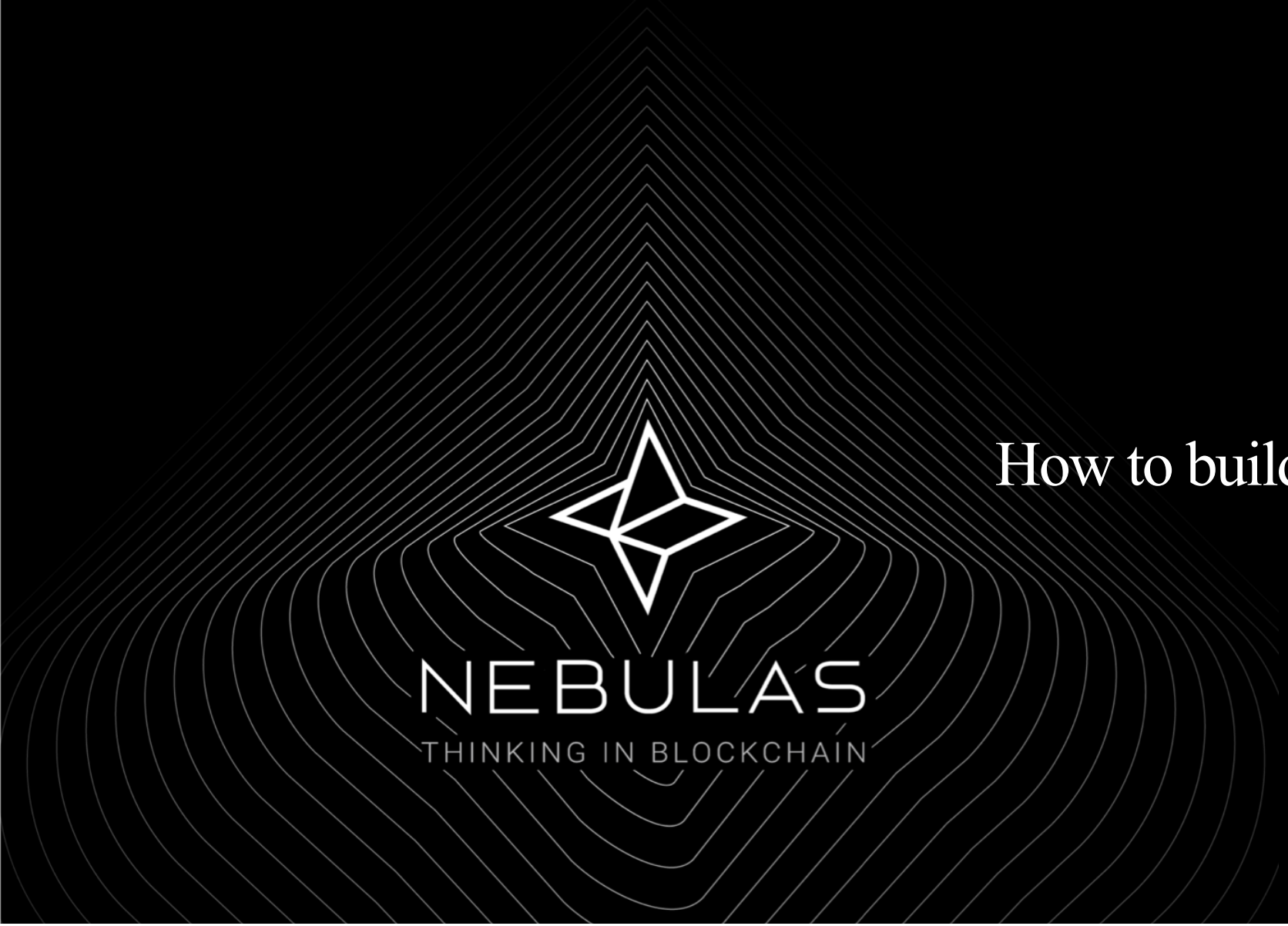


How to build DApps on Nebulas

Explore, Enjoy, Embrace

Roy Shang

NEBULAS

NEBULAS

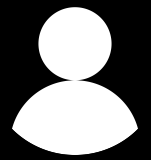
Difference between DApps and Traditional Apps

Nebulas

Manage Wallets
Create new Wallets
Unlock a Wallet
View Wallet Info

Send Transactions
Transfer Tokens
Deploy Smart Contracts
Execute Methods in Smart Contracts

Call Methods
Call Methods in Smart Contracts



Traditional Platform

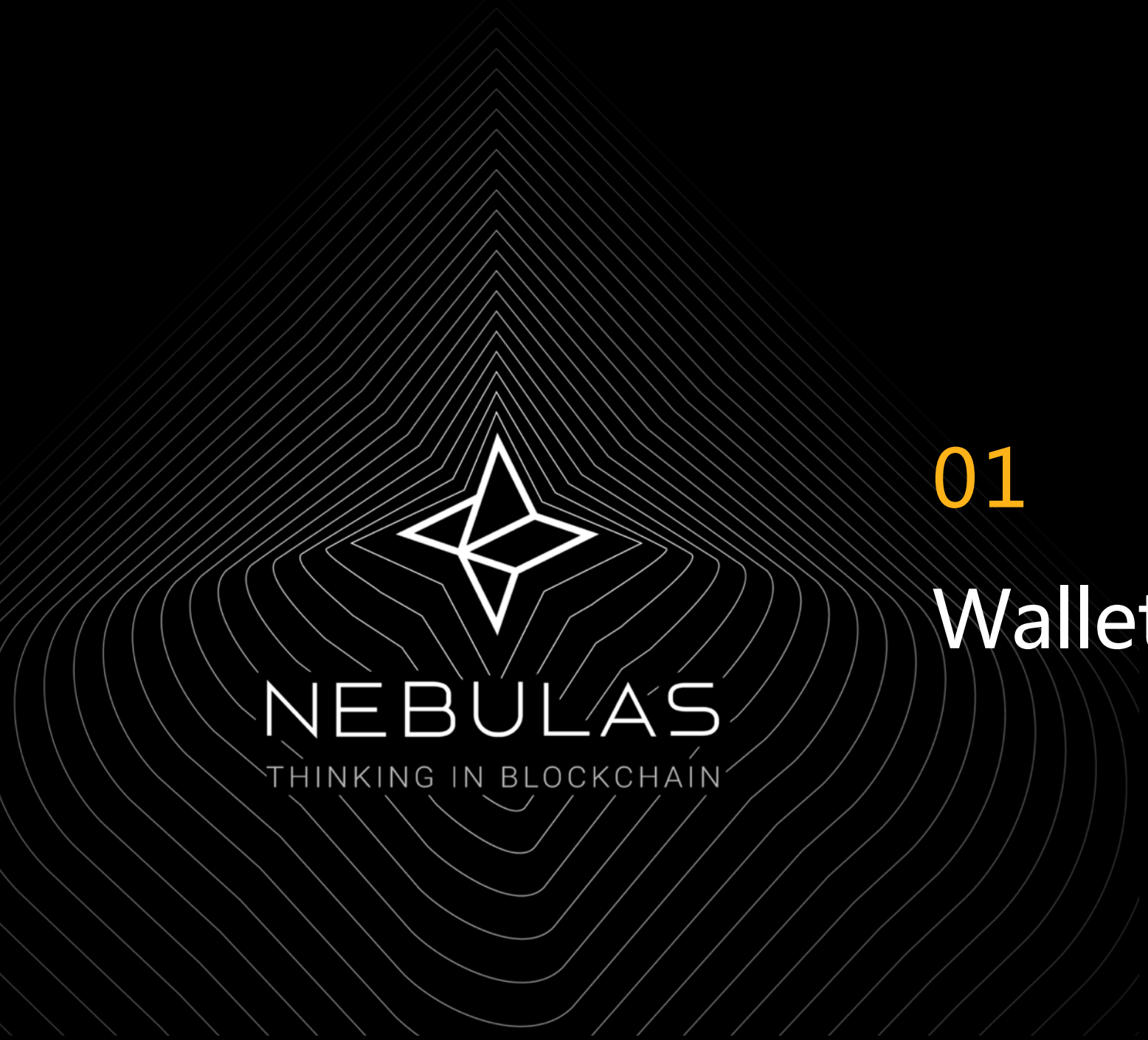
Manage Accounts
Create new Accounts
Login an Account
View Account Profile

Submit Form
Pay bills
Deploy a new Application
Stored Procedure in MySQL

Invoke Read-Only API
Fetch Data in the Application

NEBULAS

CONTENT



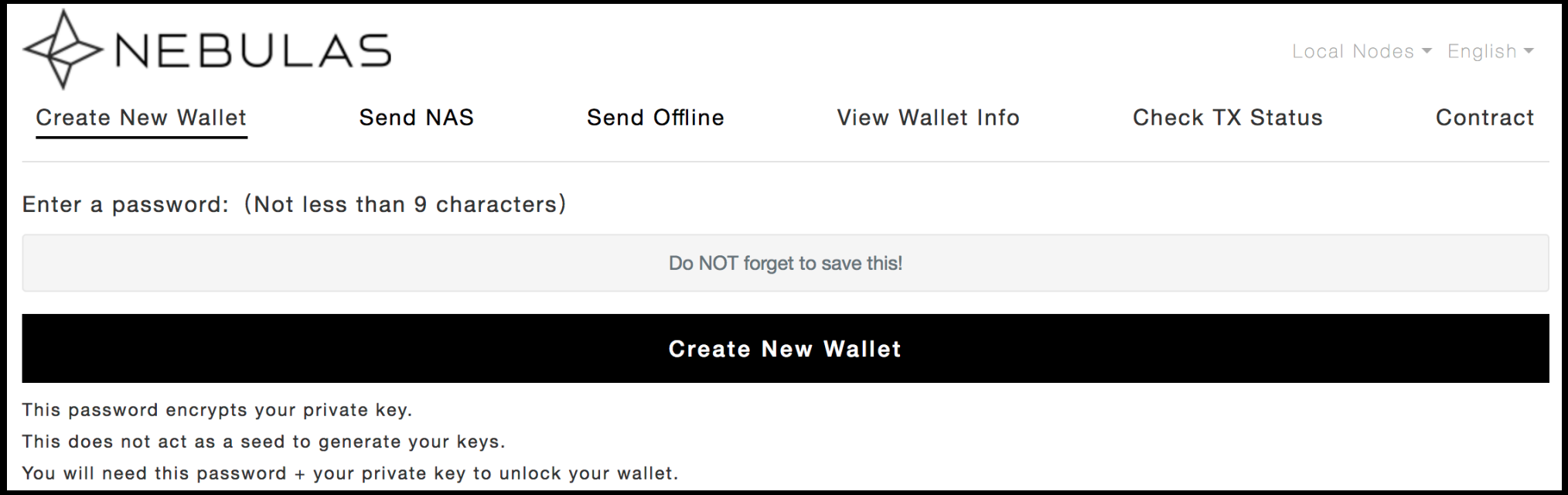
01

Wallet

NEBULAS

Create Wallet

https://github.com/nebulasio/web-wallet

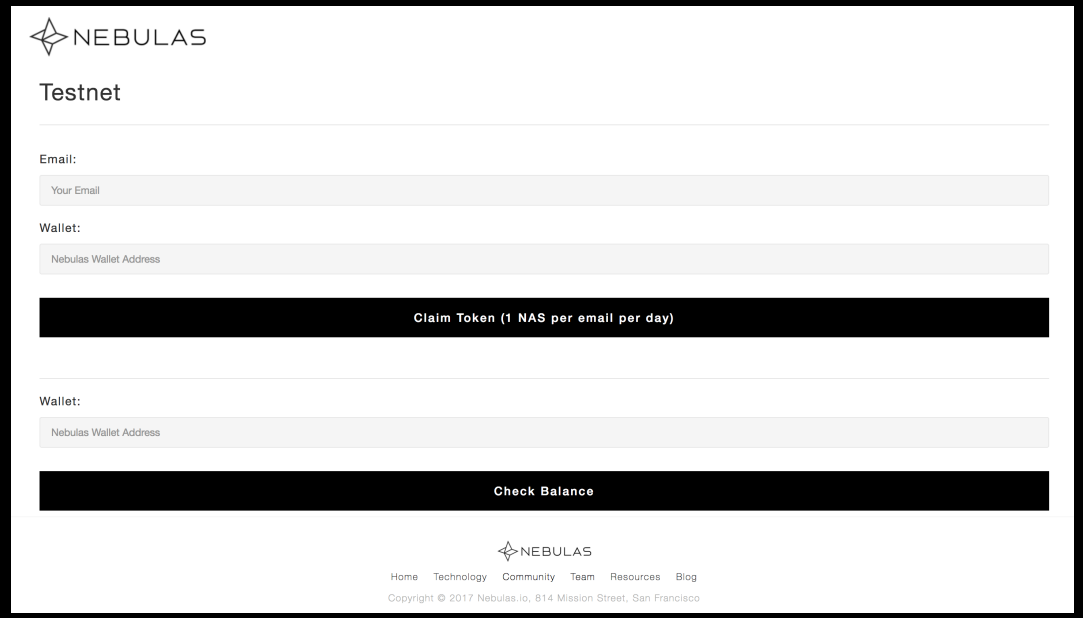


A wallet = A private key = keystore file + passphrase -> An address

NEBULAS

Prepare Tokens

Testnet: https://testnet.nebulas.io/claim/

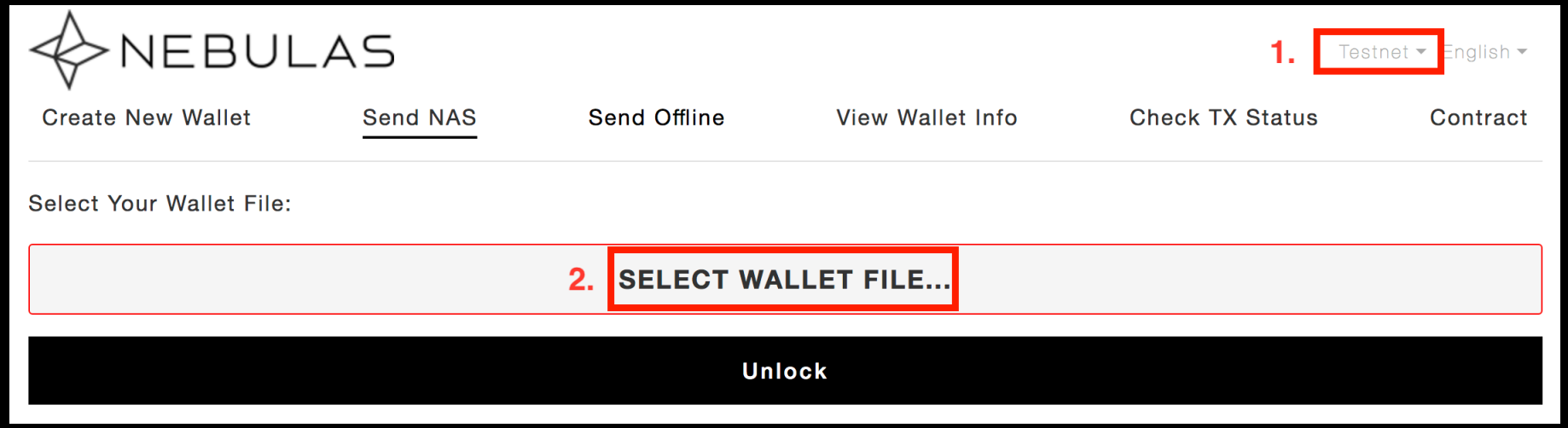


Mainnet: Exchanges

Huobi.pro

NEBULAS

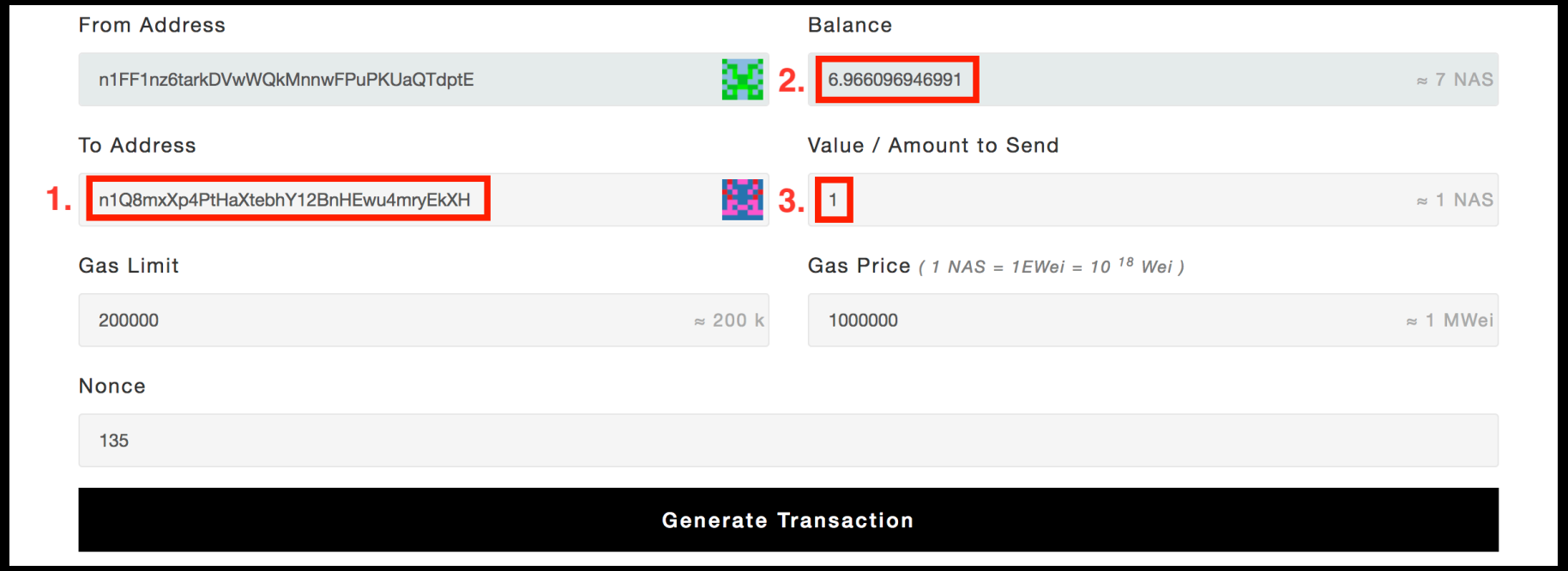
Send Transaction – Unlock Wallet



1. Select Network: Mainnet, Testnet or LocalNet
2. Import keystore file & Unlock it with passphrase

NEBULAS

Send Transaction – Arguments



1. Fill the To Address
2. Check Balance
3. Fill the Value

NEBULAS

Send Transaction – Submit

1. Transaction Arguments

2. Serialized Transaction (base64)

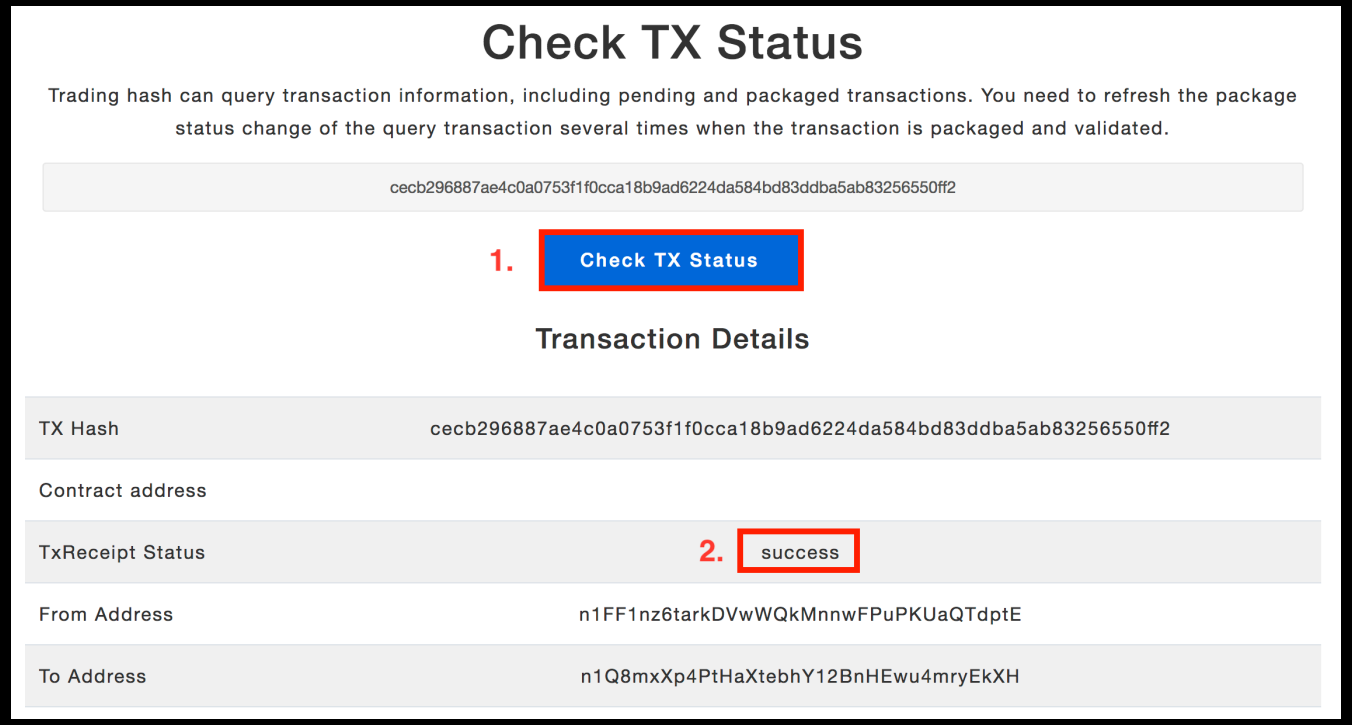
3. QR Code of Serialized Transaction

4. Click Send Transaction, wait for a while, the transaction receipt will be returned identified by txhash

5. Click txhash to check the transaction status

NEBULAS

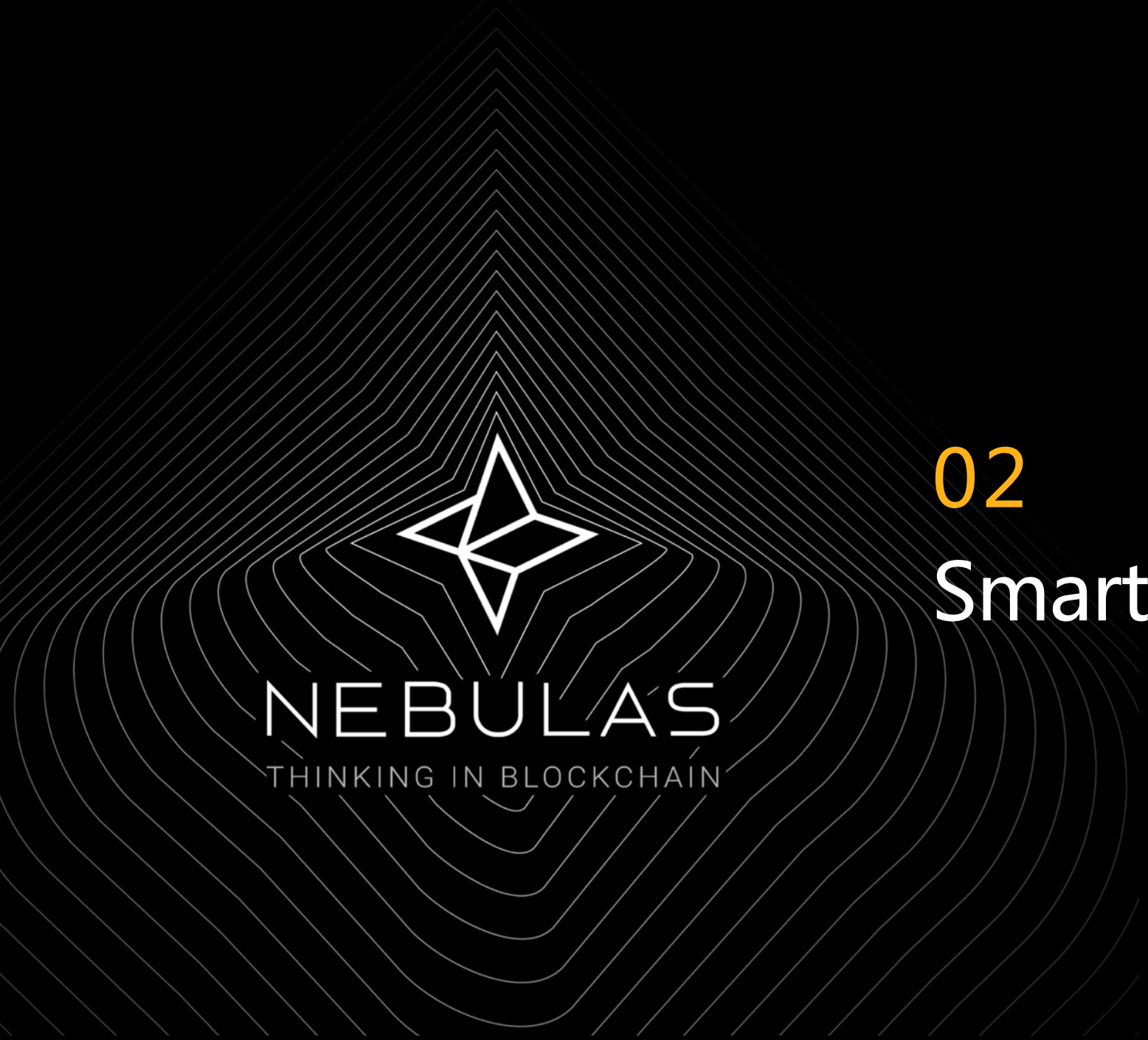
Send Transaction – Check Status



1. Refresh the status
2. In general, the status will become success after 15 seconds

If you are in the pending state for a long time

Check the balance of the sending account
Make sure it's enough to pay the gas fee



02

Smart Contracts

NEBULAS

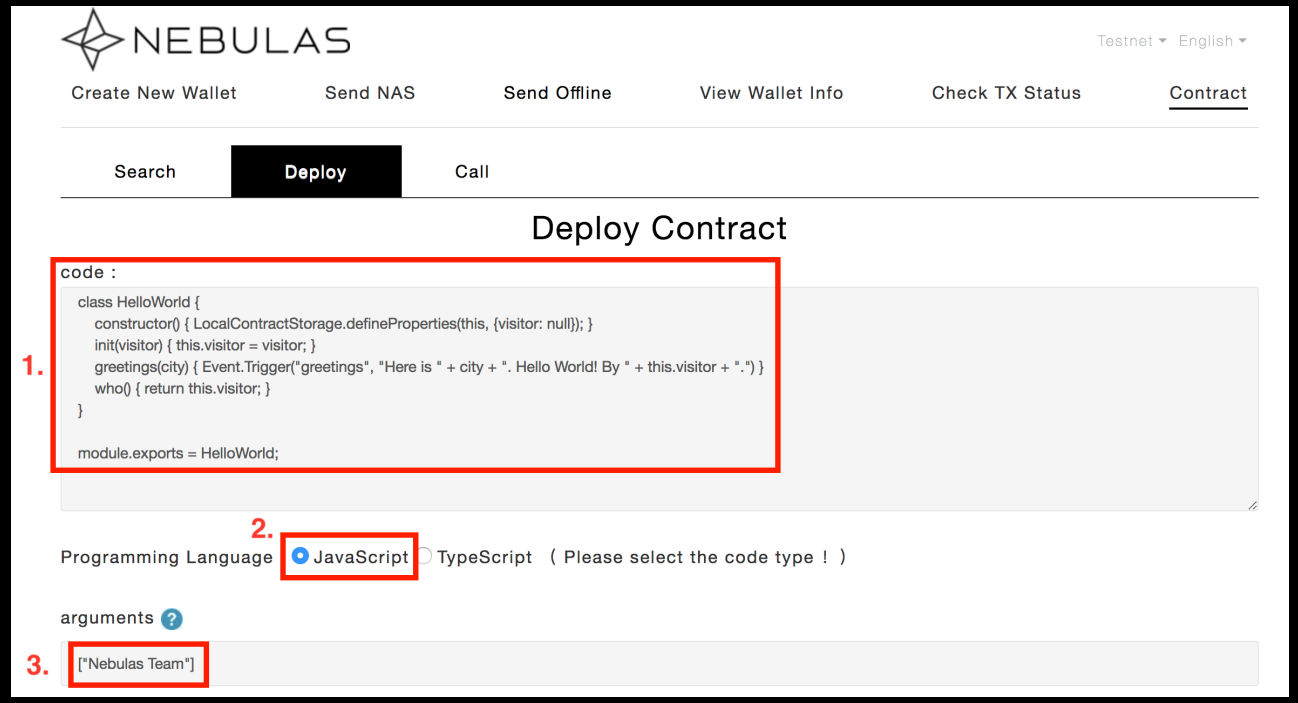
Hello World – Deploy - Write Smart Contract

```
class HelloWorld {
    constructor() {
        LocalContractStorage.defineProperties(this, {
            visitor: null
        });
    }
    init(visitor) {
        this.visitor = visitor;
    }
    greetings(city) {
        Event.Trigger("greetings", "Here is " + city + ". Hello World! By " + this.visitor + ".")
    }
    who() {
        return this.visitor;
    }
}

module.exports = HelloWorld;
```

NEBULAS

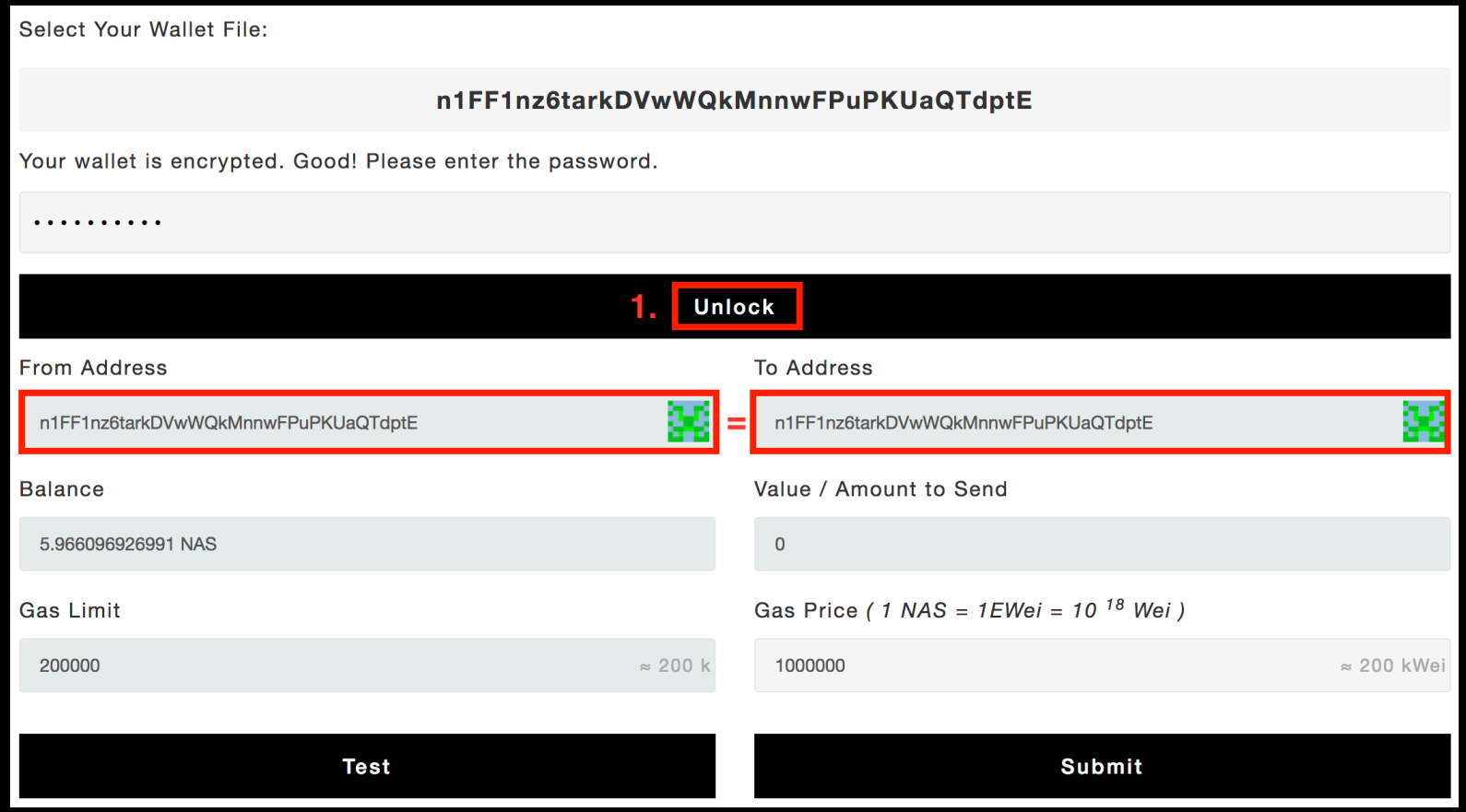
Hello World – Deploy - Write Smart Contract



1. When the contract is deployed, it is executed by default:
   - Contract constructor function: constructor
   - Contract initialization function: init
2. Select JavaScript
3. Fill in initialization parameters, array form [A, B, ...]

NEBULAS

Hello World – Deploy - Unlock Wallet



Deploy Contracts
= Send Transactions

From Address = To Address

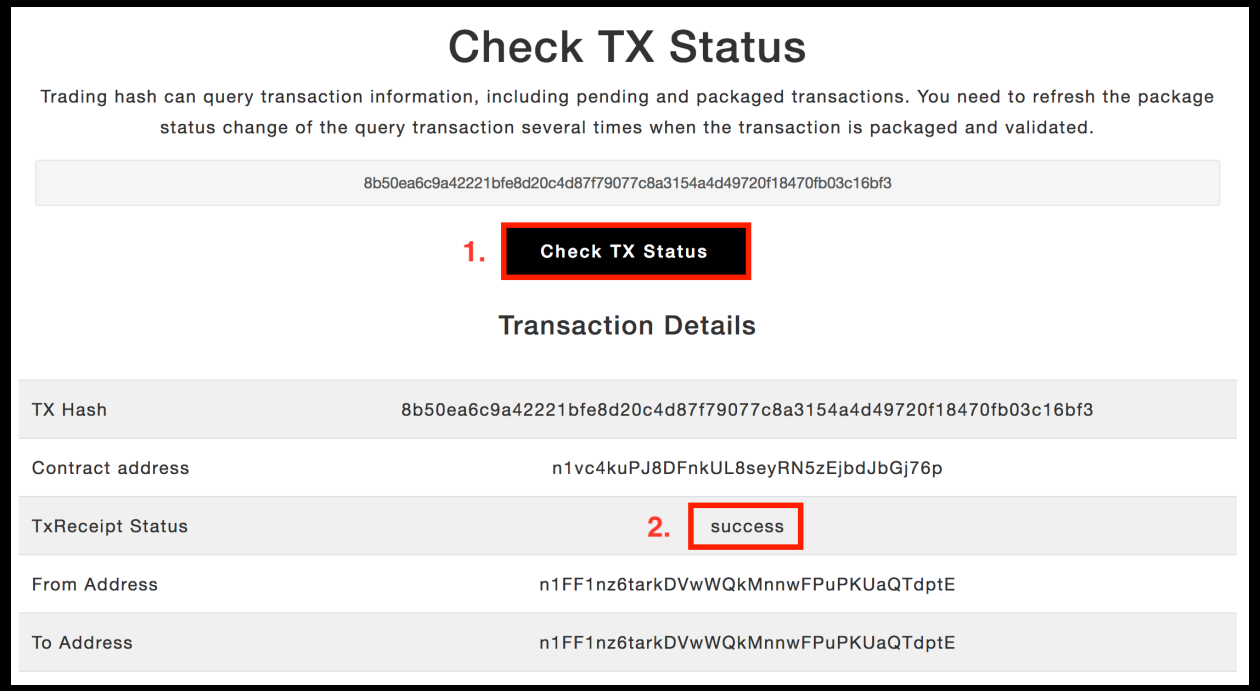
NEBULAS

Hello World – Deploy - Test & Submit

1. Test Button: Simulate the execution of constructor and init functions
2. Submit Button: Submit the contract to Nebulas platform, return txhash and contract address
3. Txhash: Click it the check the transaction status
4. Contract Address: Used as To Address in executing contracts

NEBULAS

Hello World – Deploy - Check



8b50ea6c9a42221bfe8d20c4d87f79077c8a3154a4d49720f18470fb03c16bf3

1. Refresh the status
2. In general, the status will become success after 15 seconds

If you are in the pending state for a long time

Check the balance of the sending account
Make sure it's enough to pay the gas fee

If you are in the fail state

Check the transaction using Test button again

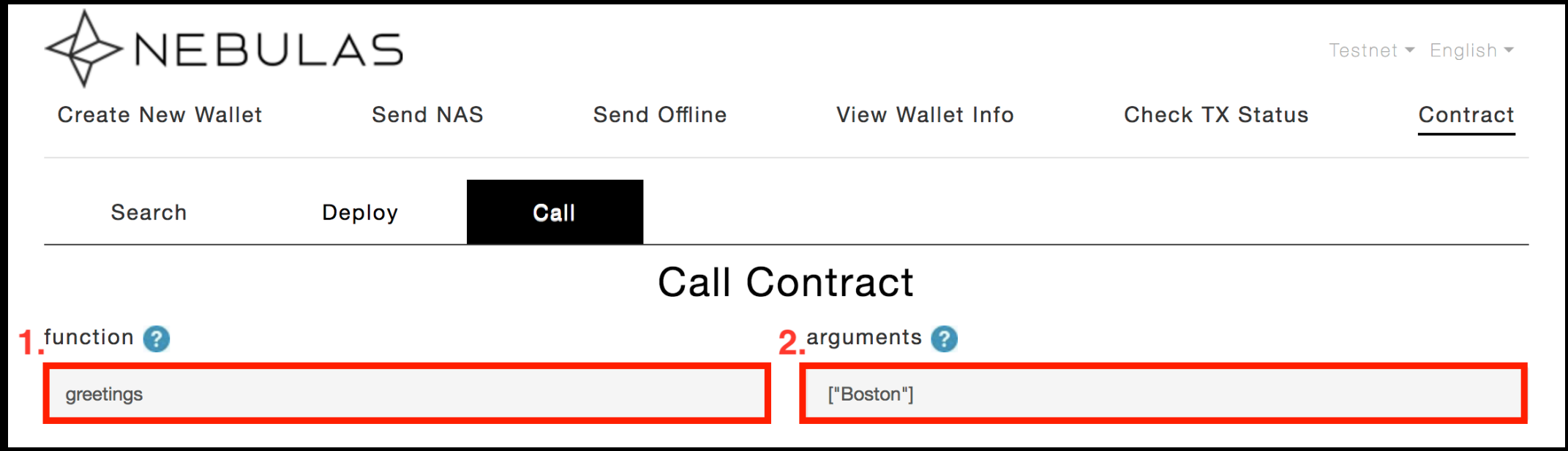
NEBULAS

Hello World – Execute - Method

1. Search contracts by deployment Txhash
2. Prepare to execute the greetings function in the contract
   - Record an event on chain

NEBULAS

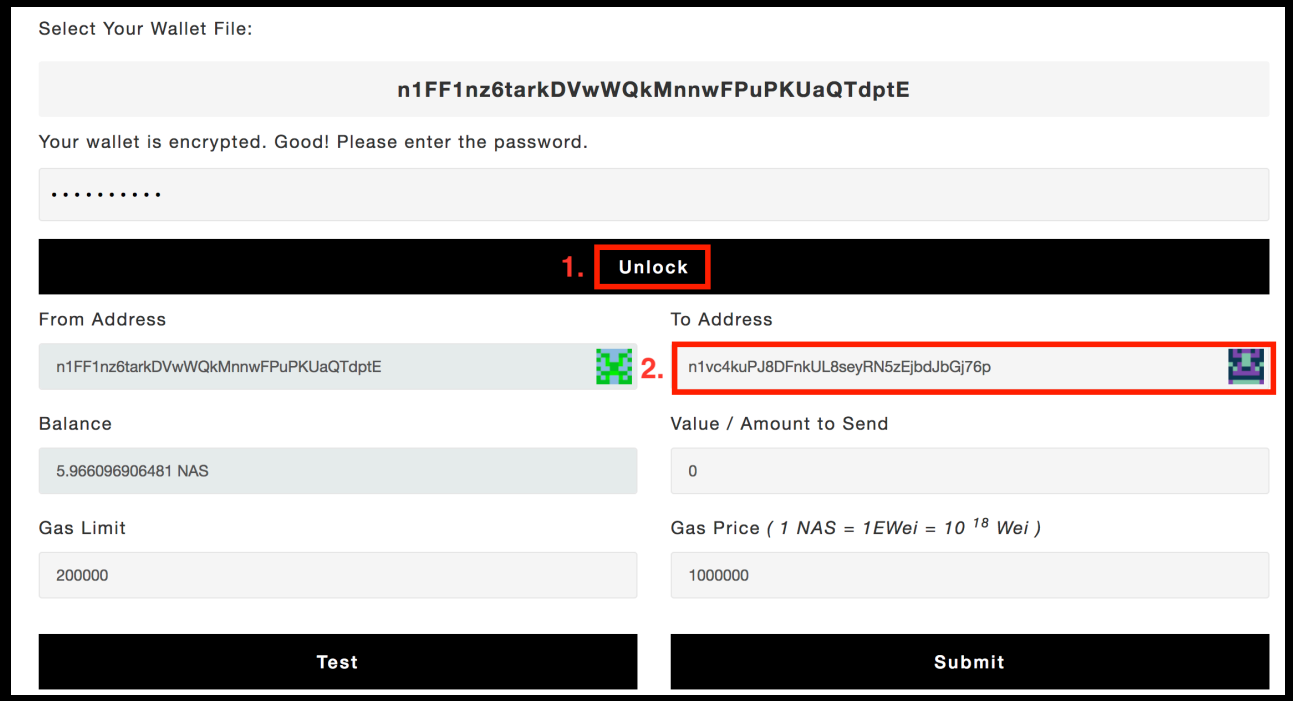
Hello World – Execute – Method Arguments



1. Fill in the function name
2. Fill in the function parameters

NEBULAS

Hello World – Execute - Contract Address

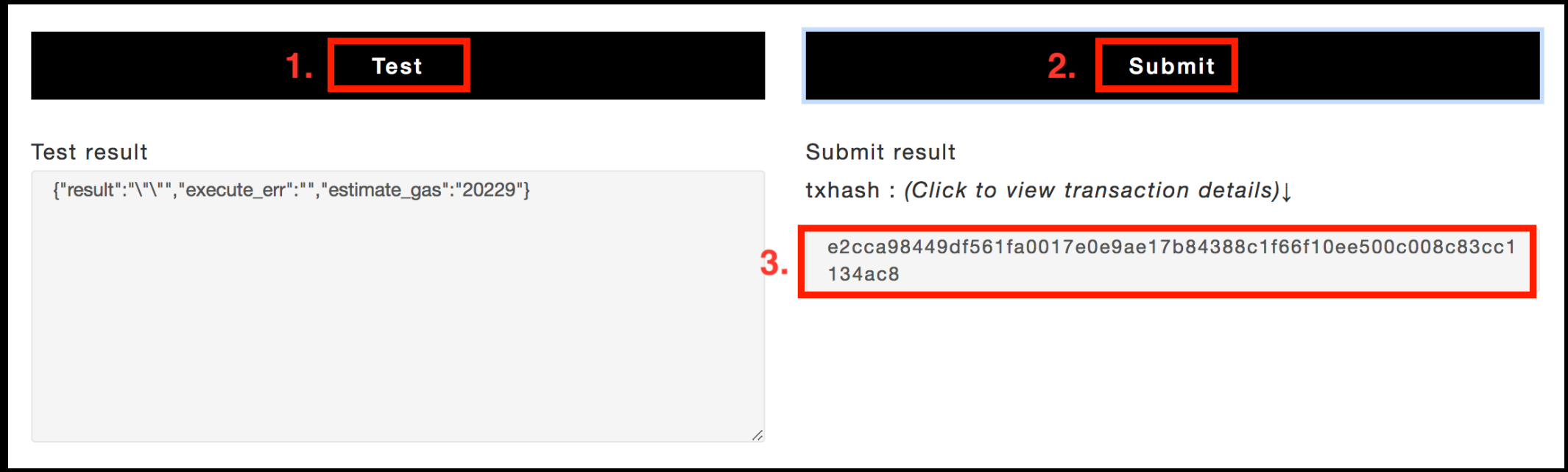


1. Unlock wallet
2. Fill in contract address

n1vc4kuPJ8DFnkUL8seyRN5zEjbdJbGj76p

NEBULAS

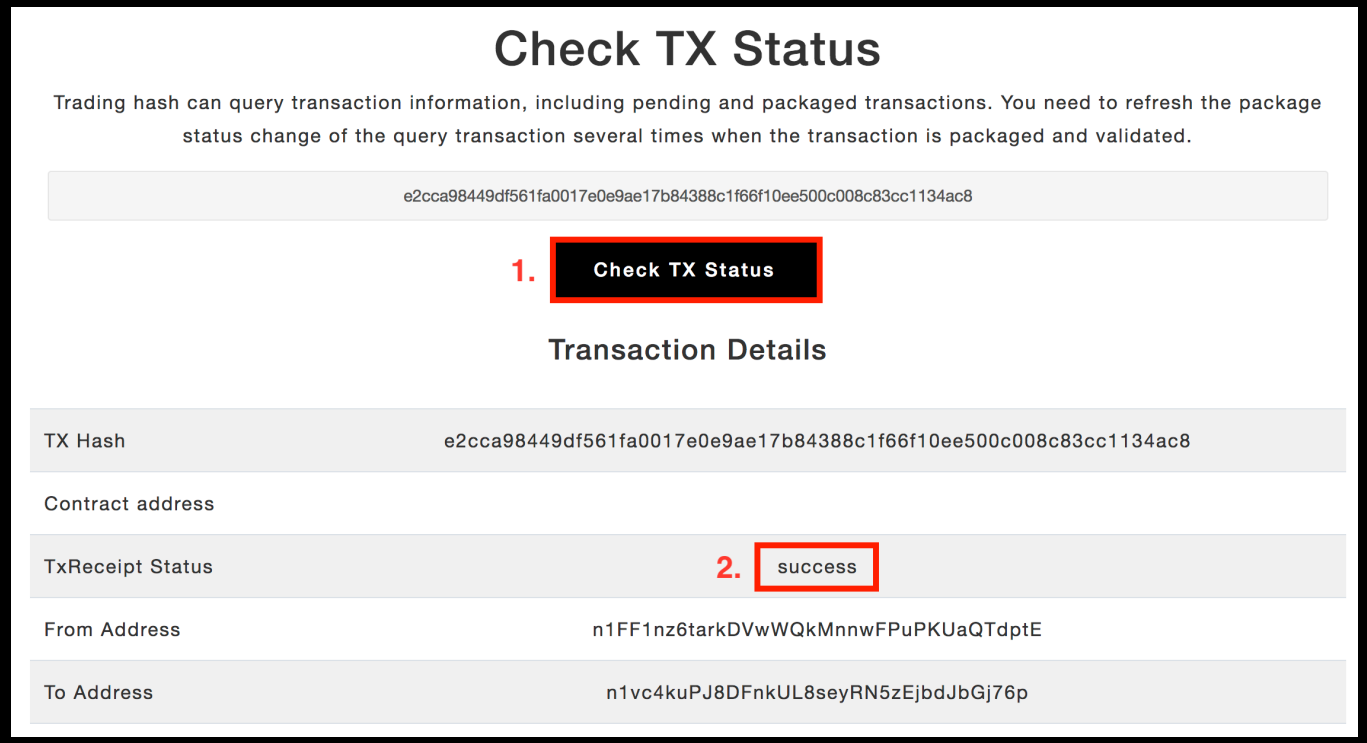
Hello World – Execute - Test & Submit



1. Test Button: Simulate the execution of greetings function
2. Submit Button: Submit the execution to Nebulas platform, return txhash
3. Txhash: Click to check the transaction status

NEBULAS

Hello World – Execute - Check



e2cca98449df561fa0017e0e9ae17b84388c1f66f10ee500c008c83cc1134ac8

1. Refresh the status
2. In general, the status will become success after 15 seconds

If you are in the pending state for a long time

Check the balance of the sending account
Make sure it's enough to pay the gas fee

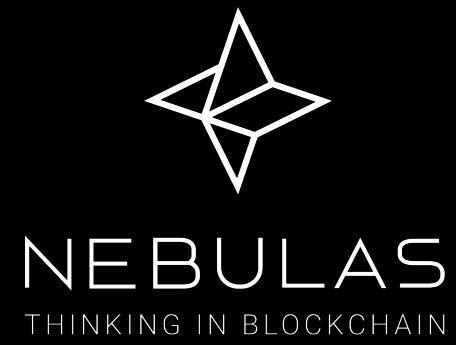
If you are in the fail state

Check the transaction using Test button again

NEBULAS

Hello World – Execute - Event

```
curl -i -H 'Content-Type: application/json' -X POST
https://testnet.nebulas.io/v1/user/getEventsByHash
-d '{"hash":"e2cc … 4ac8"}'
```




```
{
  "result":{
    "events":[
      {
        "topic": "chain.contract.greetings",
        "data":"\"Here is Boston. Hello World! By Nebulas Team.\""
      }
      ,{
        "topic":"chain.transactionResult",
        "data":"{\"hash\":\"e2cc…4ac8\",\"status\":1,\"gas_used\":\"20229\",\"error\":\"\"}"
      }
    ]
  }
}
```

NEBULAS

Hello World – Call - Methods

1. Search contracts by deployment Txhash
2. Prepare to call the who function in the contract
   - Return visitor assigned in init function

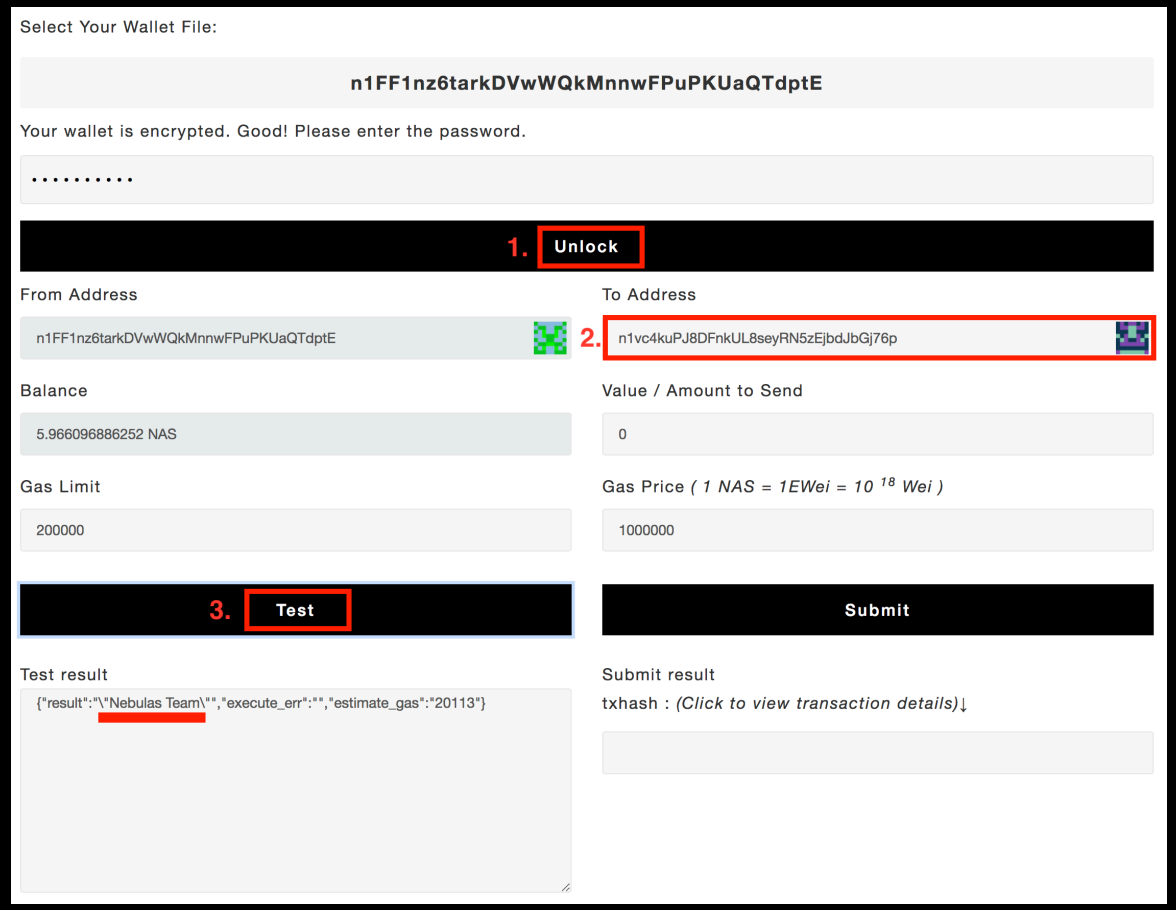
NEBULAS

Hello World – Call – Method Arguments

1. Fill in function name
2. Fill in function parameters

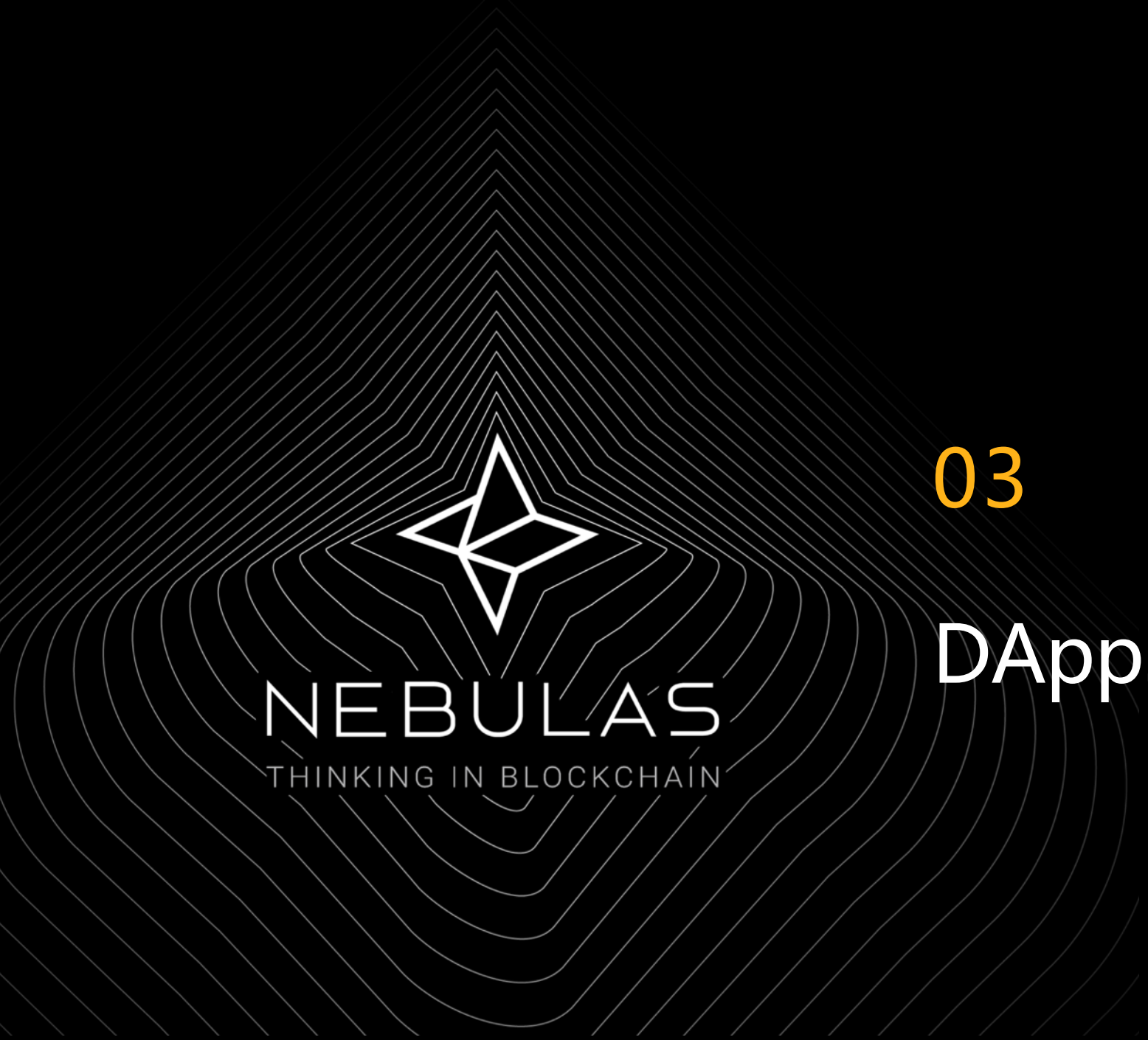
NEBULAS

Hello World – Call - Contract Address



1. Unlock wallet to get From Address
2. Fill contract address
3. Click Test button to get return value

n1vc4kuPJ8DFnkUL8seyRN5zEjbdJbGj76p



03

DApps

NEBULAS

Develop

Submit

NEBULAS

THINKING IN BLOCKCHAIN

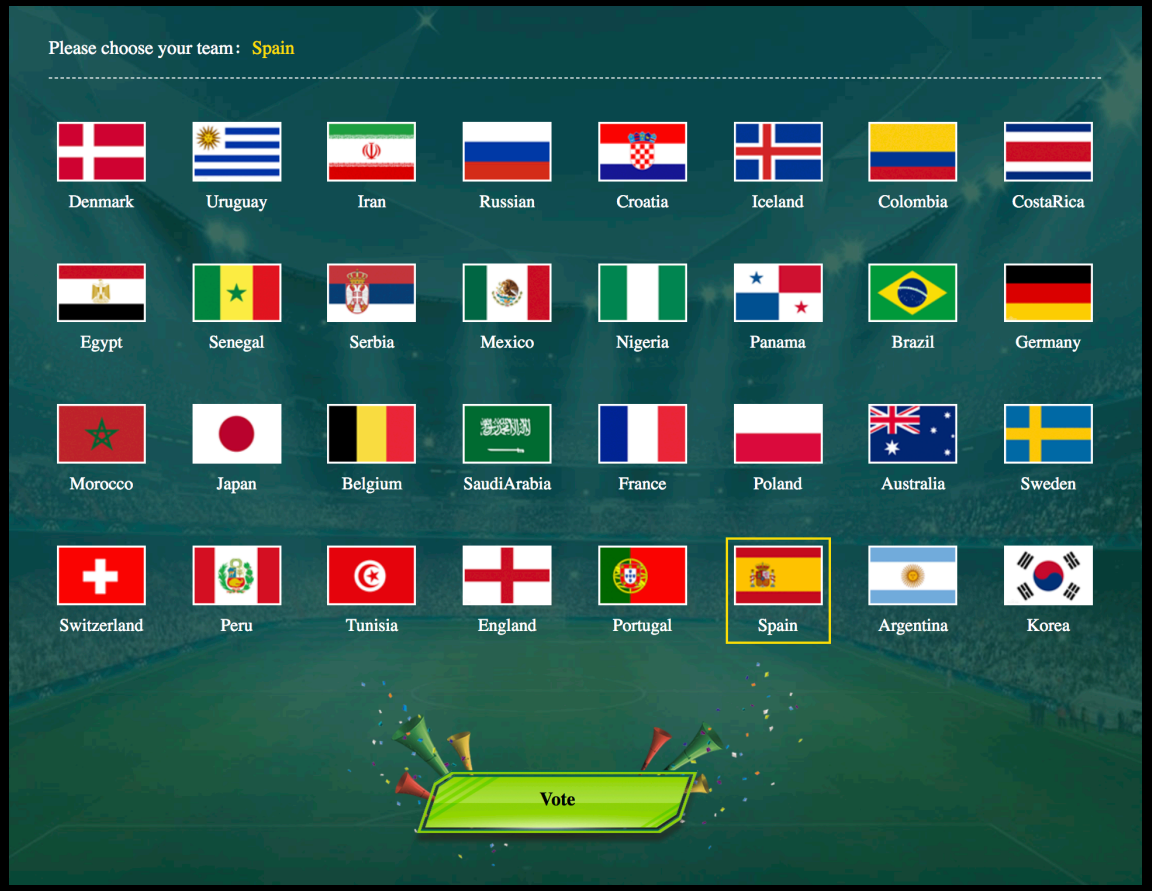
Develop

Invoke API

Interact

NEBULAS

Frontend

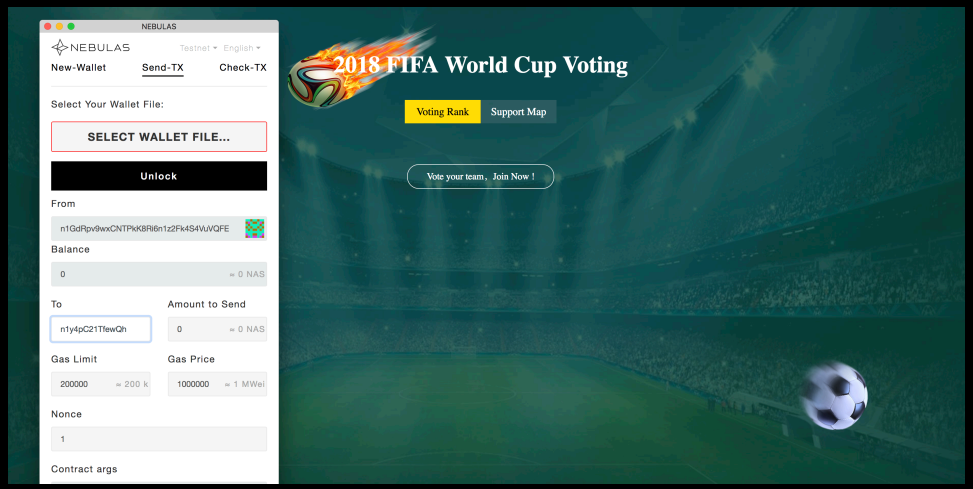


NEBULAS

User Wallets

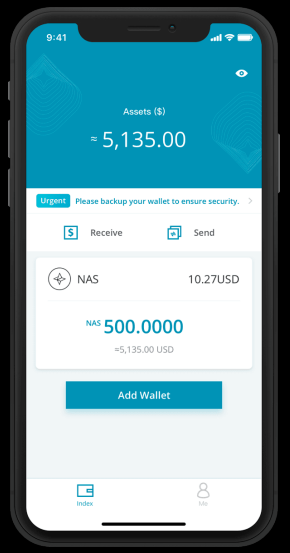
Chrome Extension – Like MetaMask in Ethereum

- https://github.com/ChengOrangeJu/WebExtensionWallet
- Support Mainnet, Testnet and Localnet
- Platform: PC



NebPay – Like ApplePay with Nebulas Mobile Wallet

- Wallet: https://nano.nebulas.io/index_en.html
- WebApp: https://github.com/nebulasio/nebPay
- IOS: https://github.com/nebulasio/neb.iOS
- Android: https://github.com/nebulasio/neb.android
- Support Mainnet
- Platform: PC & Mobile



https://github.com/nebulasio/wiki/blob/master/tutorials.md

All you need to build a Dapp can be found in this page.


nebulas.io


t.me/nebulasio


nebulasio.herokuapp.com


twitter.com/nebulasio


/r/nebulas


Channel: Nebulas


medium.com/nebulasio


contact@nebulas.io