

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Розгортання токenu SuperChainErc20 за допомогою OP Stack

Здобувач Микола ДІМІТРОВ
Група Кіукі-21-3

Керівник ас. каф.
Віталій СІТНІКОВ

Мета та задачі

2

- Метою кваліфікаційної роботи є створення, розгортання та тестування токenu за стандартом ERC-20 з додатковими функціями та задеплойти його у дві мережі OP Stack Optimism Sepolia і Base Sepolia.

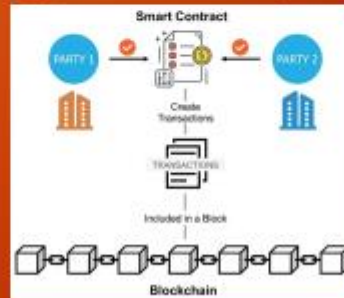
Задачі

- Аналіз сучасних фреймворків для створення смарт-контрактів.
- Реалізація та тестування токenu за стандартом ERC-20.
- Проведення деплойменту в мережі Optimism Sepolia та Base Sepolia.
- Аналіз результатів багатомережевого тестування.

Смарт-контракти

3

- Смарт-контракти – основа Web3. ERC-20 – стандарт для токенів в Ethereum. OP Stack – модульна архітектура для масштабування блокчейну.



Обрані технології

4

- Solidity v0.8.20
- OpenZeppelin
- Foundry
- Optimism та Base (Sepolia)
- RPC-провайдер Alchemy

Проектування контракту

5

- Контракт побудовано на базі OpenZeppelin ERC20, з додаванням функцій mint та burn, і захистом onlyOwner.

```

1 // OpenZeppelin Contracts (last updated v4.5.0)
2 pragma solidity ^0.8.20;
3
4 import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
5 import "@openzeppelin/contracts/access/Ownable.sol";
6
7 contract SuperToken is ERC20, Ownable {
8     constructor(uint256 initialSupply, string memory name, string memory symbol) Ownable(msg.sender) {
9         _mint(msg.sender, initialSupply * 10**decimals());
10    }
11
12    function mint(address to, uint256 amount) public onlyOwner {
13        _mint(to, amount);
14    }
15
16    function burn(uint256 amount) public {
17        _burn(msg.sender, amount);
18    }
19
20    function burnFrom(address account, uint256 amount) public onlyOwner {
21        _burn(account, amount);
22    }
23
24 }

```

Вміст файлу foundry.toml

6

```

[profile.default]
2 src = "src"
3 out = "out"
4 libs = ["lib"]
5 solc_version = "0.8.20"
6 optimizer = true
7 optimizer_runs = 200
8 via_ir = false
9 remappings = [
10     "openzeppelin/lib/openzeppelin-contracts"
11 ]
12
13 # RPC endpoints for chain of choice
14 rpc_endpoints = {}
15 optimizer_spoils = "https://spoil.spoils.io"
16 base_url = "https://api.base.org"
17 optimizer = "https://optimizer.io"
18 base = "https://base.org"
19
20 # Remappings for specific contracts in the chain
21 [etherscan]
22 optimizer_spoils = {}
23 key = "ETHERSCAN_API_KEY"
24 url = "https://api-etherscan.io/api"
25 chain = "1337"
26
27 base_spoils = {}
28 key = "BASESCAN_API_KEY"
29 url = "https://api-base.org/api"
30 chain = "8453"
31
32 optimizer = {}
33 key = "OPTIMIZER_ETHERSCAN_API_KEY"
34 url = "https://api-optimizer.io/api"
35 chain = "1"
36
37 base = {}
38 key = "BASESCAN_API_KEY"
39 url = "https://api-base.org/api"
40 chain = "8453"
41
42 # Remappings for the foundry (Fuzz) testing
43 [fuzz]

```

Перевірка для foundry.toml

7

```
root@buntu:~/superchain-erc20# forge config --basic
(profile: default)
src = "src"
out = "out"
libs = ["lib"]
remappings = [
  "openzeppelin/lib/openzeppelin-contracts/",
  "forge-std/lib/forge-std/src/",
  "openzeppelin-contracts/lib/openzeppelin-contracts/",
]
# See more config options: https://github.com/foundry-rs/foundry/blob/master/README.md#all-options
root@buntu:~/superchain-erc20#
```

Отримання тестових ЕТН

8

Суперчейн-кран
Отримайте безкоштовні тестові токени для тестування додатку на Блокчейні.

Ваш платіжний гаманець не має ідентифікації в Блокчейні!
Вказавши адресу гаманця, ви отримаєте мінімальну кількість тестових ЕТН кожні 24 години. Цей стримінг вимагає токених підтверджень з боку мережі. [Дізнатися більше](#)

Ви можете знову отримувати токени кожні 1 день, 9 годин 0 хвилин

Адреса

Мережа
 011 Cosmos Ethereum

Перевірка балансу

9

```
root@ubuntu:~/superchain-erc20# cast balance $WALLET_ADDRESS --rpc-url $OPTIMISM_SEPOLIA_RPC_URL --ether
0.025000000000000000
root@ubuntu:~/superchain-erc20# cast balance $WALLET_ADDRESS --rpc-url $BASE_SEPOLIA_RPC_URL --ether
0.025000000000000000
root@ubuntu:~/superchain-erc20#
```

Створення RPC в Alchemy для Base Sepolia

10



Розробка контракту SuperChainERC20

13

- Назва: SuperChain Token
- Символ: SCT
- Емісія: 1 000 000 SCT
- Власник: адреса розгортача

```
root@Ubuntu:~/superchain-erc20# forge build
[.] Compiling...
[.] Compiling 31 files with Solc 0.8.20
[.] Solc 0.8.20 finished in 782.32ms
Compiler run successful!
root@Ubuntu:~/superchain-erc20#
```

Тестування контракту

14

Тестування проводилося через Foundry Test Framework.

```
root@Ubuntu:~/superchain-erc20# forge test -vv
[.] Compiling...
[.] Compiling 1 files with Solc 0.8.20
[.] Solc 0.8.20 finished in 605.01ms
Compiler run successful with warnings:
Warning (2018): function state mutability can be restricted to view
   test/SuperChainERC20.t.sol:18:5:
   |
18 |     function testInitialSetup() public {
   |     ^ (Relevant source part starts here and spans across multiple lines).

Ran 5 tests for test/SuperChainERC20.t.sol:SuperChainERC20Test
[PASS] testBurn() (gas: 63014)
[PASS] testInitialSetup() (gas: 29995)
[PASS] testMint() (gas: 48997)
[PASS] testMintOnlyOwner() (gas: 15217)
[PASS] testTransfer() (gas: 45364)
Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 0.87ms (861.57µs CPU time)

Ran 2 tests for test/Counter.t.sol:CounterTest
[PASS] testFor2_setNumber(uint256) (runs: 1000, μ: 31277)
[PASS] test_Increment() (gas: 31293)
Suite result: ok. 2 passed; 0 failed; 0 skipped; finished in 22.14ms (15.51ms CPU time)

Ran 2 test suites in 27.42ms (31.01ms CPU time): 7 tests passed, 0 failed, 0 skipped (7 total tests)
root@Ubuntu:~/superchain-erc20#
```