**WOW Coin mining dApp**

**Project Overview**

The Wow Mining project is a mining contract project based on the ERC20 token. Its architecture is implemented using a scalable proxy pattern. The project consists of implementing logical contracts, Deledate; proxy contracts, Delegator; and storage variable storage contracts, Storage.

This contract allows the project party to deposit ERC20 tokens into the contract and set the starting block height for mining. When the block reaches the corresponding block height, mining begins. Each round of mining is started by the administrator, and the contract automatically calculates the total amount of mining in this round (halved). At this time, the contract operator (which can be set) is allowed to periodically extract the corresponding number of tokens from the contract to the specified address (which can be set), and generate logs. The total amount that can be mined each round will be reduced to half of the previous round's mining activity, that is, halving mining every cycle until all tokens in the contract are mined.

1. **Functional Requirements**
   1. **Roles**

* Admin: The deployment, upgrade, and control permissions of all contracts belong to the admin role. Operations such as setting tokens, setting operators, and emergency withdrawals in the contract are all performed by this role.
* Operator: The address of this role only has permission to call the withdraw() method of the contract, withdrawing tokens from the contract to the receiver address.
  1. **Features**

Wow Mining has the following features:

* Initialize： Initialize contract. (admin)
* Add or remove operator .(admin)
* Set ERC20 token before mining start.（admin）
* Set receiver .(admin)
* Set the amount of tokens mined each time (admin)
* Set the block height at which mining begins before mining starts (admin)
* Withdraw (mine) tokens from the contract to the recipient address (operator)
* Manually start the next round of mining (admin)
* Emergency withdrawal function for special situations (admin)
* Query the remaining number of tokens that can be mined in the contract (anyone)
* Query the total number of tokens already mined (anyone)
* Query tuple information for all rounds (anyone)
* Query mining information for a specific round (anyone)

**1.3 Use Cases**

1.In the creation of Wow mining contract, the following information needs to be passed: total mining amount, amount per mining attempt, starting block, ERC20 token address, and mining recipient address.

2.The administrator adds an operator through the addOperator() method.

3.After the set block height is reached, the admin initiates the next (or initial) round of mining.

4.Based on the mining progress, the operator periodically extracts a fixed amount of tokens from the contract to the recipient address.

5.After each round of mining settlement, the administrator manually initiates the next round of mining, and the above operation is repeated until the token is fully mined.

1. **Technical Requirements**

This project has been developed with Solidity language, using truffle as a development environment. Javascript is the selected language for testing and scripting.

In addition，OpenZeppelin’s libraries are used in the project. All information about the contracts library and how to install it can be found in their Github.

├── **contracts**

│   ├── **test**

│   │   └── **MockToken.sol**

│   └── **wow**

│   ├── **WowMiningPoolDelegate.sol**

│   ├── **WowMiningPoolDelegator.sol**

│   └── **WowMiningPoolStorage.sol**

├── **coverage**

├── **migrations**

├── **package.json**

├── **test**

│   └── **WowMining.test.js**

├── **truffle-config.js**

└── **yarn.lock**

The main smart contracts for Wow mining are located in the ./contracts/wow folder. WowMiningPoolStorage.sol contains the state variables of the contract, which includes two smart contracts: WowMiningPoolStorage and WowMiningPoolAdminStorage. WowMiningPoolDelegator.sol is a proxy contract primarily used to control the logic upgrade of the contract, while WowMiningPoolDelegate.sol is a delegate contract primarily used to write the business logic of the contract.

In the ./test folder, WowMining.test.js provides unit tests for various contract methods. The command "truffle test ./test/**WowMining.test.js**" can be used to run the unit tests.

The project configuration information can be found in ./**truffle-config.js**, which includes the names and version information of some dependent modules. For specific configurations, please refer to Truffle's official documentation.

For specific contract deployment or method usage, please refer to the contract's unit tests. A contract has already been deployed on the BSC test network:

https://testnet.bscscan.com/address/0x2D8b16c69EF9dDADF4db11843705ca35c856429A#readProxyContract