

oceanico.io



SMART CONTRACTS EXPERTISE —
RIGHT WAY TO SUCCESS
HOWDOO Smart Contract Auditt

If you have any questions concerning
smart contract design and audit, feel
free to contact zoia@oceanico.io

Content

Description of the set of procedures for auditing a smart contract	3
Terms of Reference for the creation of a smart contract	4
List of audited files	7
Review of smart contract #1	8
Results of contract audit	12

Description of the complex of procedures for auditing a smart contract

1. Primary architecture review

- Checking the architecture of the contract.
- The correctness of the code.
- Checking for linearity, shortness, and self-documentation.
- Static verification and code analysis for validity and the presence of syntactic errors.

2. Comparison of requirements and implementation

- Checking the code of the smart contract for compliance with the requirements of the customer code logic, writing algorithms, matching the initial constant values.
- Identification of potential vulnerabilities

3. Testing according to the requirements

- Control testing of a smart contract for compliance with specified customer requirements.
- Running tests of the properties of the smart contract in test net.

Terms of Reference for the creation of a smart contract

MultivestPro	Private Presale \$0.05	Public Presale Tier 1 \$0.08	Public ICO Tier 1 \$0.09	Public ICO Tier 2 \$0.10
--------------	---------------------------	---------------------------------	-----------------------------	-----------------------------

TIMING

Start Date	Flexible	02.05.2018	12.06.2018	Right after Tier 1 ends
Start Time		09.59 AM	11.59 AM	
Timezone		GMT	GMT	
End Date	01.05.2018	02.06.2018	12.07.2018	12.07.2018
End Time	11.59 PM	11.59 PM	11.59 PM	11.59 PM
Timezone	GMT	GMT	GMT	GMT

[x] Max balance before end triggers next phase or end of ICO

Comments whitelisting of SC investor's address, owner receives payment in fiat and sends tokens to the investor
if all tokens sold, set the SC on hold until next tier, SC owner can update start date of public ICO manually

THRESHOLDS

Indicated in:

[x] tokens issued

x] contributions made (indicate currency) ETH,BTC,LTC

Decimals 18

Soft Cap No

Hard Cap 888.888.888,00

Min per transaction 250 USD flexibly changeable

Max per transaction No

Calculation unit (ETH, token, USD) USD

TOKEN

Token Name	uDOO	
Token Symbol	uDOO	
Total for sale supply	311.111.110,80	
Total to raise	888.888.888,00	
Token Price	different tiers	
Token Preminted [Yes No]	No	sent to 2nd party on demand through function on Multivest
Refund [Yes No]	No	
Burn Tokens [Yes No]	No	
Irreversible [Yes No]	No	
Available to contributor right after individual contribution [Yes No]	Yes	
Instantly transferable [Yes No]	No	
Instantly enable allow / transfer-From [Yes No]	No	

TIER

Pre-Allocate	88.888.888,80	multiple addresses	users are gonna claim
Private Sale	44,444,444.40	@ 0,05 USD	
Private Sale Round 2 Pre-ICO	44,444,444.40	@ 0,09 USD	
Public Presale	133,333,333.20	@ 0,08 USD	
Public ICO Tier 1	88,888,888.80	@ 0,09 USD	
Public ICO Tier 2	44.444.444,40	@ 0,10 USD	
Public ICO Tier 3	44.444.444,40	@ 0,17 USD	

CONTRIBUTIONS

Immediately withdrawable [Yes No]	Yes	immediately sent to an address
--	-----	--------------------------------

TOKEN DISTRIBUTION

Dividends

[Yes No]	No
--------------	----

Allocation

Team/Vested Tokens:	101.210.666,57	Allocation after the start of the ICO
Brett King	3.733.333,33	12 months, 25% released every 3 month
Paul Mears	3.111.111,11	12 months, 25% released every 3 month
Ian Gilmore	777.777,78	12 months, 25% released every 3 month
Andy Hones	3.733.333,33	12 months, 25% released every 3 month
Mark Perring	808.888,89	36 months, 2.77% each month.
Neil Harper	964.444,44	36 months, 2.77% each month.
Tony Loan	149.333,33	36 months, 2.77% each month.
Nakul Sha	1.066.666,67	12 months, 25% released every 3 month.
David Brierley	62.222.222,16	12 months, 25% released every 3 month.
Treasury tokens:	177,777,777.60	transferred instantly once the contract has been deployed on the mainnet
ETH address		Where the ETH goes to re contributions

ETH Owners address

Airdrop

[Yes No]	Yes
--------------	-----

Any token not allocated in the Token rounds, 50% will be airdropped to existing contributors and 50% will be moved over to the HIS.

Howdoo Incentive Scheme	191,111,110.92	transferred instantly once the contract has been deployed on the mainnet
Referral Tokens	10 000 000	Sent to 1 address, transferrable
Whitelist	21,111,111.08	
[Yes No]	yes	

List of audited files

Github:

<https://github.com/Howdoo-tech/contracts/commit/e586655bd47c39881cc065af47f2c8853af04043>

The experts conducted an audit of 24 .sol files on the list:

- SafeMathMock.sol
- StandardTokenMock.sol
- TestHowdoo.sol
- TestHowdooAllocation.sol
- TestICO.sol
- TestMultivest.sol
- TestPrivateSale.sol
- BasicToken.sol
- ERC20.sol
- ERC20Basic.sol
- Howdoo.sol
- HowdooAllocation.sol
- HowdooERC20.sol
- ICO.sol
- Migrations.sol
- MintingERC20.sol
- Multivest.sol
- OraclizeAPI.sol
- Ownable.sol
- PrivateSale.sol
- Referral.sol
- SafeMath.sol
- SellableToken.sol
- StandardToken.sol

Review of smart contract #1

Howdoo Smart Contract Review

<https://github.com/Howdoo-tech/contracts/commit/e586655bd47c39881cc065af47f2c8853af04043>

Important

1. Possible bugs

- 1.1. No critical issues found.

General

1. We recommend to update the `pragma solidity` to the current version.
2. We recommend using the `zeppelin-solidity` library and connecting with it the contracts ERC20, ERC20Basic, BasicToken, Ownable, SafeMath, StandardToken.
3. `ethereumjs-testrpc` is deprecated, we recommend using `ganache-cli`.
4. The function `calculateTokensAmount` is declared as constant but calls non-constant functions <https://github.com/Howdoo-tech/contracts/blob/master/contracts/ICO.sol#L105>.
5. The function `calculateEthersAmount` is declared as constant but calls non-constant functions.
6. We recommend to bring all the listed checks to a single format: <https://github.com/Howdoo-tech/contracts/blob/master/contracts/HowdooERC20.sol#L51>
<https://github.com/Howdoo-tech/contracts/blob/master/contracts/HowdooERC20.sol#L56-L58>
<https://github.com/Howdoo-tech/contracts/blob/master/contracts/HowdooERC20.sol#L63-L65>

Review of smart contract #1

<https://github.com/Howdoo-tech/contracts/blob/master/contracts/HowdooERC20.sol#L70-L72>

<https://github.com/Howdoo-tech/contracts/blob/master/contracts/HowdooERC20.sol#L77-L79>

We recommend using `require(!locked)` or returning `false` in all cases. When using `require(!locked)`, you can create a modifier `notLocked()` and use it instead of checking. The best solution for `locked` implementation, in our experience, is the use of `PausableToken` from `zeppelin-solidity`. <https://github.com/OpenZeppelin/zeppelin-solidity/blob/master/contracts/token/ERC20/PausableToken.sol>

7. Part of the tests of the contract does not pass

node v8.10.0

Truffle v4.1.5

solcjs 0.4.19+commit.c4cbbb05.Emscripten.clang

Ganache CLI v6.1.0 (ganache-core: 2.1.0)

```
46 passing (2m)
2 failing
```

- 1) Contract: ICO setEtherInUSD & isPreICOActive & isICOFinished:
`AssertionError: isPreICOActive not equal: expected true to equal false`
at Context.<anonymous> (test/ico.js:490:16)
at <anonymous>
at process._tickCallback (internal/process/next_tick.js:188:7)
- 2) Contract: PrivateSale check moveUnsoldTokens:
`AssertionError: maxAmount is not equal: expected undefined to equal '1.77753879e+26'`
at Context.<anonymous> (test/private_sale.js:546:16)
at <anonymous>
at process._tickCallback (internal/process/next_tick.js:188:7)

Gas optimization

1. The variable `disableMinting` is not used anywhere. We recommend to remove it for gas optimization. <https://github.com/Howdoo-tech/contracts/blob/master/contracts/MintingERC20.sol#L18>.
2. We recommend to remove this method for gas optimization <https://github.com/Howdoo-tech/contracts/blob/master/contracts/Referral.sol#L52-L56>.
3. The variable is not used. We recommend to remove it for gas optimization. <https://github.com/Howdoo-tech/contracts/blob/master/contracts/Multivest.sol#L12>

Review of smart contract #1

4. Don't need to transfer to the `_address` function, if it should always be equal to `msg.sender` <https://github.com/Howdoo-tech/contracts/blob/master/contracts/Referral.sol#L42>. We recommend to remove this test to optimize the gas.
5. In line 28, `maxSupply` is initialized, and then a new `maxSupply` assignment is performed on line 32. We recommend doing this in one place to optimize the gas. <https://github.com/Howdoo-tech/contracts/blob/master/contracts/Howdoo.sol#L22-L35>

Code quality

1. Cyclic relationships between components (contracts) greatly complicate the understanding of the work of contracts. We recommend revising the architecture of the project as a whole.
2. Here we recommend writing just `require(allowedMultivests[_address])` without explicit comparison with `true`. <https://github.com/Howdoo-tech/contracts/blob/master/contracts/Multivest.sol#L24>
3. In this check, we recommend to replace with `require(whitelist[msg.sender]&&buy(msg.sender,msg.value))` <https://github.com/Howdoo-tech/contracts/blob/master/contracts/PrivateSale.sol#L44>
4. The names of the functions `isPreICOActive` and `isICOFinished` symbolize that functions perform only checking without any side effects, although they change the state of the contract.
5. This function does not have enough indents <https://github.com/Howdoo-tech/contracts/blob/master/contracts/HowdooAllocation.sol#L66-L136>
6. Do not recommend sent `now` to `allocateInternal`. The same value can be obtained inside this method by calling `now`. The value `now` does not change during the whole transaction. <https://github.com/Howdoo-tech/contracts/blob/master/contracts/HowdooAllocation.sol#L138>,
7. We recommend to replace the `mint` function <https://github.com/Howdoo-tech/contracts/blob/master/contracts/MintingERC20.sol#L50-L68> by <https://gist.github.com/e7097c1fc331a11b2c710e0a85c2ee0c>.
8. The return value of `uint256` is not used anywhere except for comparison with the original value of `_amount` inside `require.require` <https://github.com/Howdoo-tech/>

Review of smart contract #1

[contracts/blob/master/contracts/SellableToken.sol#L171-L173](https://github.com/Howdoo-tech/contracts/blob/master/contracts/SellableToken.sol#L171-L173) this can be slightly simplified by replacing with `require(howdoo.mint (_address, _tokenAmount))`

9. We recommend to use `PRE_ICO_ID` instead of `i` <https://github.com/Howdoo-tech/contracts/blob/master/contracts/ICO.sol#L101>. We don't recommend use variables before they are declared.
10. The similar construction we recommend that you replace with `minInvest.mul(1ether).div (etherPriceInUSD)`. In most cases, these wrappers are not needed. <https://github.com/Howdoo-tech/contracts/blob/master/contracts/ICO.sol#L194>
11. We recommend using the `pure` and `view` functions that appeared in the `solidity 0.4.16` to replace the `constant` keyword in the signatures. <https://github.com/Howdoo-tech/contracts/blob/master/contracts/ICO.sol#L94>
12. We recommend using `async / await` in tests to not use `then` <https://github.com/Howdoo-tech/contracts/blob/master/test/erc20.js#L70-L94>. This test would look like this <https://gist.github.com/kolya-t/5112b742c75f05a655bb34d4030991e4>.
13. In `web3.js` there are functions `toWei` and `fromWei` <https://github.com/ethereum/wiki/wiki/JavaScript-API#web3toWei>, they are very convenient to use. A similar construction can be replaced using `web3.toWei`. <https://github.com/Howdoo-tech/contracts/blob/master/test/ico.js#L165>

Results of contract audit

<https://github.com/Howdoo-tech/contracts/commit/e586655bd47c39881cc065af47f2c8853af04043>

The information in this report is a list of recommendations that need to be followed to ensure the quality and safety of the smart contract.

The experts audited the contract. Based on the results, the developers of the smart contract were given recommendations for improving the optimization of the smart contract code.

The current version of the code did not reveal any critical issues.

For all questions regarding the audit and testing of the smart contract, we recommend contacting zoia@oceanico.io