Smart Contract
Security Audit
V1

# Umoja Coin Smart Contract Audit

Aug 5, 2025

Audited By: SaferICO

# Table of Contents

## Background

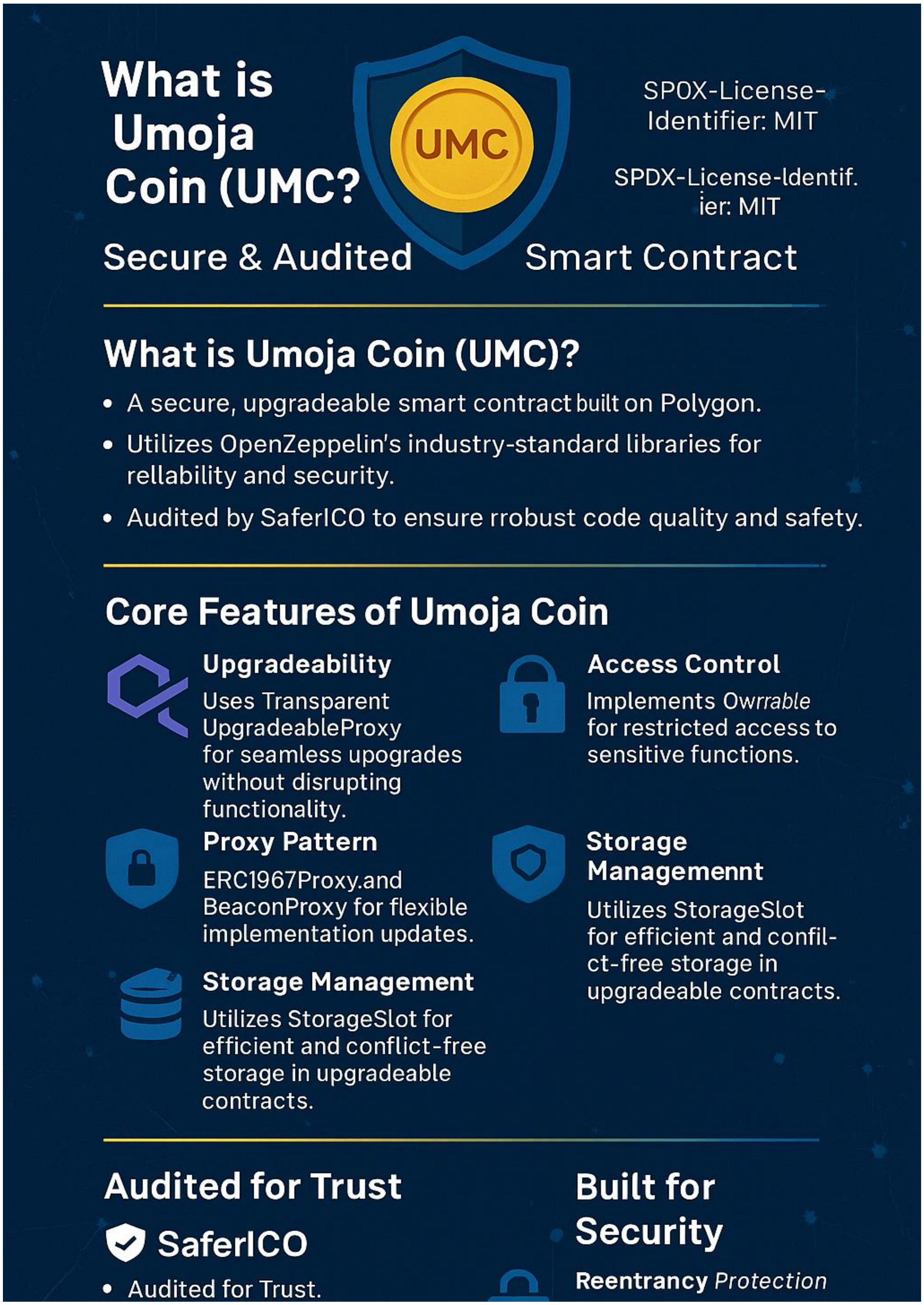
The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

## Project Information

- **Platform**: Polygon Chain
- **Name**: Umoja Coin (UMC)
- **Language** : Solidity
- **Contract Address**: 0x80f2c9ed338bfce2bb128eccbb9b11bbca041a82, 0xc6fd60ecd78d5cdf9efb15b2a50b1411b2fe4c62
- **Code Source**: https://polygonscan.com/address/0x80f2c9ed338bfce2bb128eccbb9b11bbca041a82#code https://polygonscan.com/address/0xc6fd60ecd78d5cdf9efb15b2a50b1411b2fe4c62#code



# What is Umoja Coin (UMC?

UMC

SP0X-License-Identifier: MIT

SPDX-License-ldentif. ier: MIT

Secure & Audited Smart Contract

## What is Umoja Coin (UMC)?

- A secure, upgradeable smart contract built on Polygon.
- Utilizes OpenZeppelin's industry-standard libraries for rellability and security.
- Audited by SaferICO to ensure rrobust code quality and safety.

## Core Features of Umoja Coin

**Upgradeability**
Uses Transparent UpgradeableProxy for seamless upogrades without disrupting functionality.

**Proxy Pattern**
ERC1967Proxy.and BeaconProxy for flexible implementation updates.

**Storage Management**
Utilizes StorageSlot for efficient and conflict-free storage in upgradeable contracts.

**Access Control**
Implements Ow*rrable* for restricted access to sensitive functions.

**Storage Managemennt**
Utilizes StorageSlot for efficient and confil-ct-free storage in upgradeable contracts.

## Audited for Trust

SaferICO

- Audited for Trust.

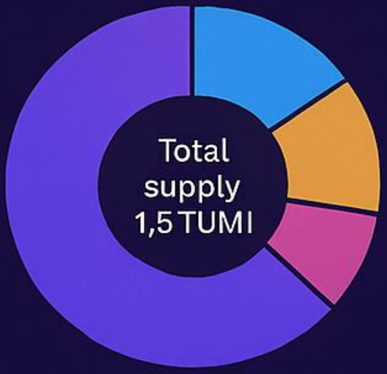
## Built for Security

**Reentrancy** *Protection*

## Core Details

- **Token Name:** UmojaCoin (UMC)
- **Standard:** ERC20 (Upgradeable
- **License:** MIT
- **Solidity Version:** ^.8.0
- **Features:** Upgradeable (UUPS), Access Control, Vesting Schedules

## Token Distribution



| | | | |
|---|---|---|---|
| ● | Early Stage | 10 M UMC | 6.67% |
| ● | Presale | 100 M UMC | 6.67% |
| ● | Co-founders | 291,25 M UMC | 15.75% |
| ● | Developers | 250 M UMC | 13.67% |
| ● | Marketing | 66.3 M UMC | 4.42% |
| ● | Free for Trading: | 792.45 MUM | 52.65% |

## Vesting Schedules

- 12-Month Vesting: 1% on listing, 1% monthly for 6 months, remaining over 6 months
- 24-Month Vesting: 1% on listing, 1% monthly for 6 months, remaining over 18 months
- Key function: releaseVestedTo-kenst) Formula:

  Vested amount calculated based on elapsed time, duration, and category

## Key Addresses & Permissions

- Treasury: 0x99eb...02b8
- Marketing: 0x5bb3...FD4b
- Developer: 0x4a0b...DaD9

**Admin Role**
Blacklist acccounts. allocate tokens, wifr and funds, upgrade contract

**Blacklist**
Prevents token transfers for blacklisted accounts

## Core Functionalities

- Initialize
- Allocate Tokens (Admin only)
- Release Vested Tokens

## Tracked Events

- Tokens Purchased
- Tokens Vested

## Executive Summary

According to our assessment, the customer`s solidity smart contract is **Well-Secured**.

| | |
|---|---|
| Well Secured | ✓ |
| **Secured** | |
| Poor Secured | |
| Insecure | |

Automated checks are with remix IDE. All issues were performed by the team, which included the analysis of code functionality, manual audit found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the audit overview section. The general overview is presented in the Project Information section and all issues found are located in the audit overview section.

Team found 0 critical, 0 high, 0 medium, 2 low, 0 very low-level issues and 2 notes in all solidity files of the contract

The files:

UMC.sol

## Audit Score:

99% secure

## File and Function Level Report

### File in Scope:

| Contract Name | SHA 256 hash | Contract Address |
|---|---|---|
| UMC.sol | 5d2ba2062d2d6d4d8306bedb0996931195154ec1 | 0x80f2c9ed338bfce2bb128eccbb9b11bbca041a82 0xc6fd60ecd78d5cdf9efb15b2a50b1411b2fe4c62 |

- Contract: ERC1967Proxy
- Inherit: Proxy
- Observation: All passed including security check
- Test Report: passed
- Score: passed
- Conclusion: passed

| Function | Test Result | Type / Return Type | Score |
|---|---|---|---|
| allowance | ✓ | Read / public | Passed |
| ADMIN_ROLE | ✓ | Read / public | Passed |
| blacklisted | ✓ | Read / public | Passed |
| hasRole | ✓ | Read / public | Passed |
| balanceOf | ✓ | Read / public | Passed |
| decimals | ✓ | Read / public | Passed |
| name | ✓ | Read / public | Passed |
| symbol | ✓ | Read / public | Passed |
| calculateVestedAmount | ✓ | Read / public | Passed |
| COFOUNDERS_TOKENS | ✓ | Read / public | Passed |
| DEFAULT_ADMIN_ROLE | ✓ | Read / public | Passed |
| EARLY_STAGE_TOKENS | ✓ | Read / public | Passed |
| DEVELOPERS_TOKENS | ✓ | Read / public | Passed |

| | | | |
|---|---|---|---|
| DEVELOPER_WALLET | ✓ | Read / public | **Passed** |
| FREE_FOR_TRADING_TOKENS | ✓ | Read / public | **Passed** |
| getRoleAdmin | ✓ | Read / public | **Passed** |
| getVestingDuration | ✓ | Read / public | **Passed** |
| proxiableUUID | ✓ | Read / public | **Passed** |
| listing_time | ✓ | Read / public | **Passed** |
| MARKETING_TOKENS | ✓ | Read / public | **Passed** |
| MARKETING_WALLET | ✓ | Read / public | **Passed** |
| PRESALE_TOKENS | ✓ | Read / public | **Passed** |
| supportsInterface | ✓ | Read / public | **Passed** |
| TREASURY_WALLET | ✓ | Read / public | **Passed** |
| TOTAL_SUPPLY | ✓ | Read / public | **Passed** |
| totalSupply | ✓ | Read / public | **Passed** |
| UPGRADE_INTERFACE_VERSION | ✓ | Read / public | **Passed** |
| vestingSchedules | ✓ | Read / public | **Passed** |
| approve | ✓ | Write / public | **Passed** |
| transfer | ✓ | Write / public | **Passed** |
| transferFrom | ✓ | Write / public | **Passed** |
| allocateTokens | ✓ | Write / public | **Passed** |
| initalize | ✓ | Write / public | **Passed** |
| renounceRole | ✓ | Write / public | **Passed** |
| revokeRole | ✓ | Write / public | **Passed** |
| setBlacelist | ✓ | Write / public | **Passed** |
| setListingTime | ✓ | Write / public | **Passed** |
| withdrawFunds | ✓ | Write / public | **Passed** |
| withdrawTokens | ✓ | Write / public | **Passed** |
| releaseVestedTokens | ✓ | Write / public | **Passed** |

| | | | |
|---|---|---|---|
| upgradeToAndCall | ✓ | Write / payable | **Passed** |
| decreaseAllowance | ✓ | Write / public | **Passed** |
| upgradeTo | ✓ | Write / public | **Passed** |
| increaseAllowance | ✓ | Write / public | **Passed** |
| grantRole | ✓ | Write / public | **Passed** |

# Issues Checking Status

## SWC Attack Analysis

The Smart Contract Weakness Classification Registry (SWC Registry) is an implementation of the weakness classification scheme proposed in EIP-1470. It is loosely aligned to the terminologies and structure used in the Common Weakness Enumeration (CWE) for more info check https://swcregistry.io/

| No. | Issue Description | Checking Status |
|---|---|---|
| 136 | Unencrypted Private Data On-Chain | Passed |
| 135 | Code With No Effects | Passed |
| 134 | Message call with hardcoded gas amount | Passed |
| 133 | Hash Collisions With Multiple Variable Length Arguments | Passed |
| 132 | Unexpected Ether balance | Passed |
| 131 | Presence of unused variables | Passed |
| 130 | Right-To-Left-Override control character (U+202E) | Passed |
| 129 | Typographical Error | Passed |
| 128 | DoS with block gas limit. | Passed |
| 127 | Arbitrary Jump with Function Type Variable | Passed |
| 126 | Insufficient Gas Griefing | Passed |
| 125 | Incorrect Inheritance Order | Passed |
| 124 | Write to Arbitrary Storage Location | Passed |
| 123 | Requirement Violation | Passed |
| 122 | Lack of Proper Signature Verification | Passed |
| 121 | Missing Protection against Signature Replay Attacks | Passed |
| 120 | Weak Sources of Randomness from Chain Attributes | Passed |
| 119 | Shadowing State Variables | Passed |

| | | |
|---|---|---|
| 118 | Incorrect Constructor Name | **Passed** |
| 117 | Signature Malleability | **Passed** |
| 116 | Block values as a proxy for time | **Not Passed** |
| 115 | Authorization through tx.origin | **Passed** |
| 114 | Transaction Order Dependence | **Passed** |
| 113 | DoS with Failed Call | **Passed** |
| 112 | Delegatecall to Untrusted Callee | **Passed** |
| 111 | Use of Deprecated Solidity Functions | **Passed** |
| 110 | Assert Violation | **Passed** |
| 109 | Uninitialized Storage Pointer | **Passed** |
| 108 | State Variable Default Visibility | **Passed** |
| 107 | Reentrancy | **Passed** |
| 106 | Unprotected SELFDESTRUCT Instruction | **Passed** |
| 105 | Unprotected Ether Withdrawal | **Passed** |
| 104 | Unchecked Call Return Value | **Passed** |
| 103 | Floating Pragma | **Not Passed** |
| 102 | Outdated Compiler Version | **Passed** |
| 101 | Integer Overflow and Underflow | **Passed** |
| 100 | Function Default Visibility | **Passed** |

## Severity Definitions

| Risk Level | Description |
|---|---|
| Critical | Critical vulnerabilities are usually straightforward to exploit and can lead to tokens loss etc. |
| High | High-level vulnerabilities are difficult to exploit; however, they also have significant impact on smart contract execution,<br>e.g. public access to crucial functions |
| Medium | Medium-level vulnerabilities are important to fix; however, they can’t lead to tokens lose |
| Low | Low-level vulnerabilities are mostly related to outdated, unused etc. code snippets, that can’t have significant impact on execution |
| Note | Lowest-level vulnerabilities, code style violations and info statements can’t affect smart contract execution and can be ignored. |

## Audit Findings

### Critical:

No Critical severity vulnerabilities were found.

### High:

No High severity vulnerabilities were found.

### Medium:

No Medium severity vulnerabilities were found.

### Low:

**# Hardcoded Wallet Addresses**

Description

Limits flexibility. Future deployments can't change this unless the code is redeployed.

```
address public constant TREASURY_WALLET = ...;
```

Recommendation

Pass wallet addresses via `initialize()` and store them in `immutable` or `private/internal` state variables with a setter if truly needed.

Status: Acknowledged.

**#Missing Receive/Fallback Function**

Description

You call withdrawFunds using address(this).balance, but the contract has no way to **receive ETH** unless it's via a token swap.

Remediation

```
receive() external payable {}.
```

### Very Low:

No Very Low severity vulnerabilities were found.

**Notes:**

#Pragam version not fixed

Description

It is a good practice to lock the solidity version for a live deployment (use 0.8.30 instead of ^0.8.20). contracts should be deployed with the same compiler version and flags that they have been tested the most with. Locking the pragma helps ensure that contracts do not accidentally get deployed using, for example, the latest compiler which may have higher risks of undiscovered bugs. Contracts may also be deployed by others and the pragma indicates the compiler version intended by the original authors. And avoid Solidity compiler Bugs check here

https://sepolia.etherscan.io/solcbuginfo

Remediation

Remove the ^ sign to lock the pragma version.

Use of block.timestamp for comparisons

The value of block.timestamp can be manipulated by the miner. And conditions with strict equality is difficult to achieve - block.timestamp.

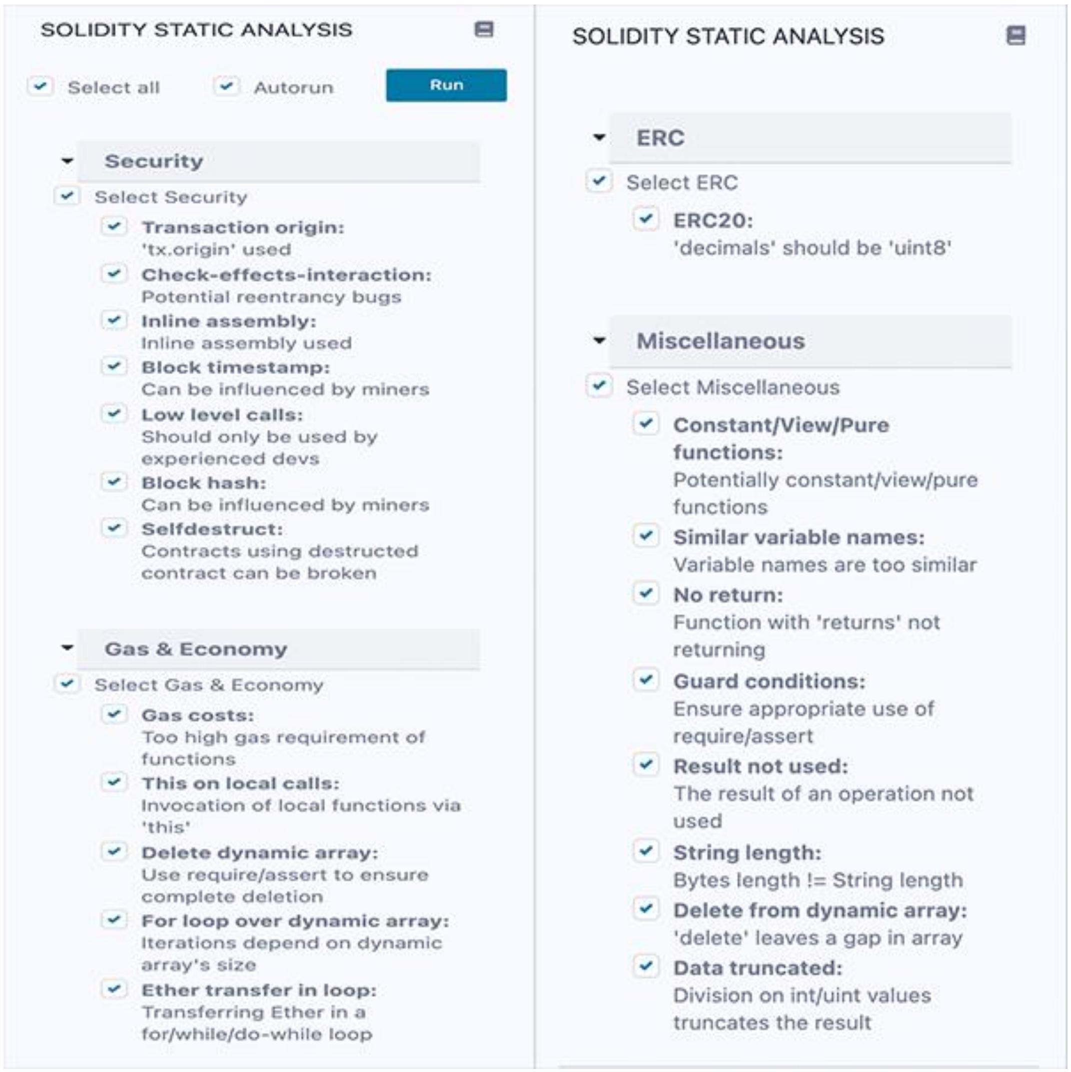
```
function setListingTime(uint256 _listingTime) external onlyAdmin {
        require(
            _listingTime > block.timestamp,
            "Listing time must be in the future"
        );
        listingTime = _listingTime;
    }

    function releaseVestedTokens() external
notBlacklisted(msg.sender) {
        require(listingTime > 0, "Listing time not set");
        require(listingTime < block.timestamp, "Vesting not started
yet");
```

Recommendation

Avoid use of block.timestamp.

# Automatic Testing

## 1- SOLIDITY STATIC ANALYSIS



SOLIDITY STATIC ANALYSIS

✓ Select all ✓ Autorun Run

**Security**

✓ Select Security

- ✓ **Transaction origin:** 'tx.origin' used
- ✓ **Check-effects-interaction:** Potential reentrancy bugs
- ✓ **Inline assembly:** Inline assembly used
- ✓ **Block timestamp:** Can be influenced by miners
- ✓ **Low level calls:** Should only be used by experienced devs
- ✓ **Block hash:** Can be influenced by miners
- ✓ **Selfdestruct:** Contracts using destructed contract can be broken

**Gas & Economy**

✓ Select Gas & Economy

- ✓ **Gas costs:** Too high gas requirement of functions
- ✓ **This on local calls:** Invocation of local functions via 'this'
- ✓ **Delete dynamic array:** Use require/assert to ensure complete deletion
- ✓ **For loop over dynamic array:** Iterations depend on dynamic array's size
- ✓ **Ether transfer in loop:** Transferring Ether in a for/while/do-while loop

SOLIDITY STATIC ANALYSIS

**ERC**

✓ Select ERC

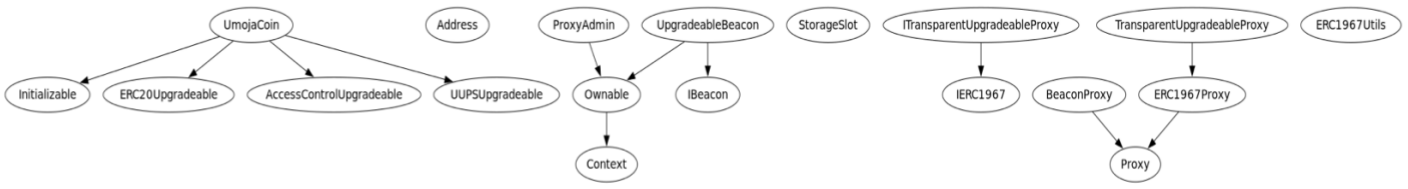
- ✓ **ERC20:** 'decimals' should be 'uint8'

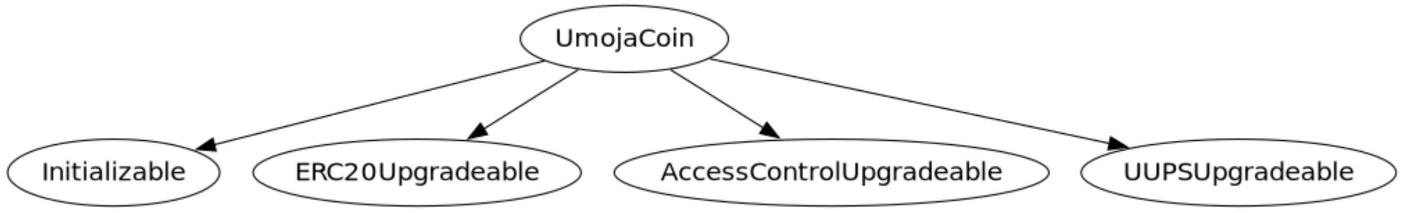
**Miscellaneous**

✓ Select Miscellaneous

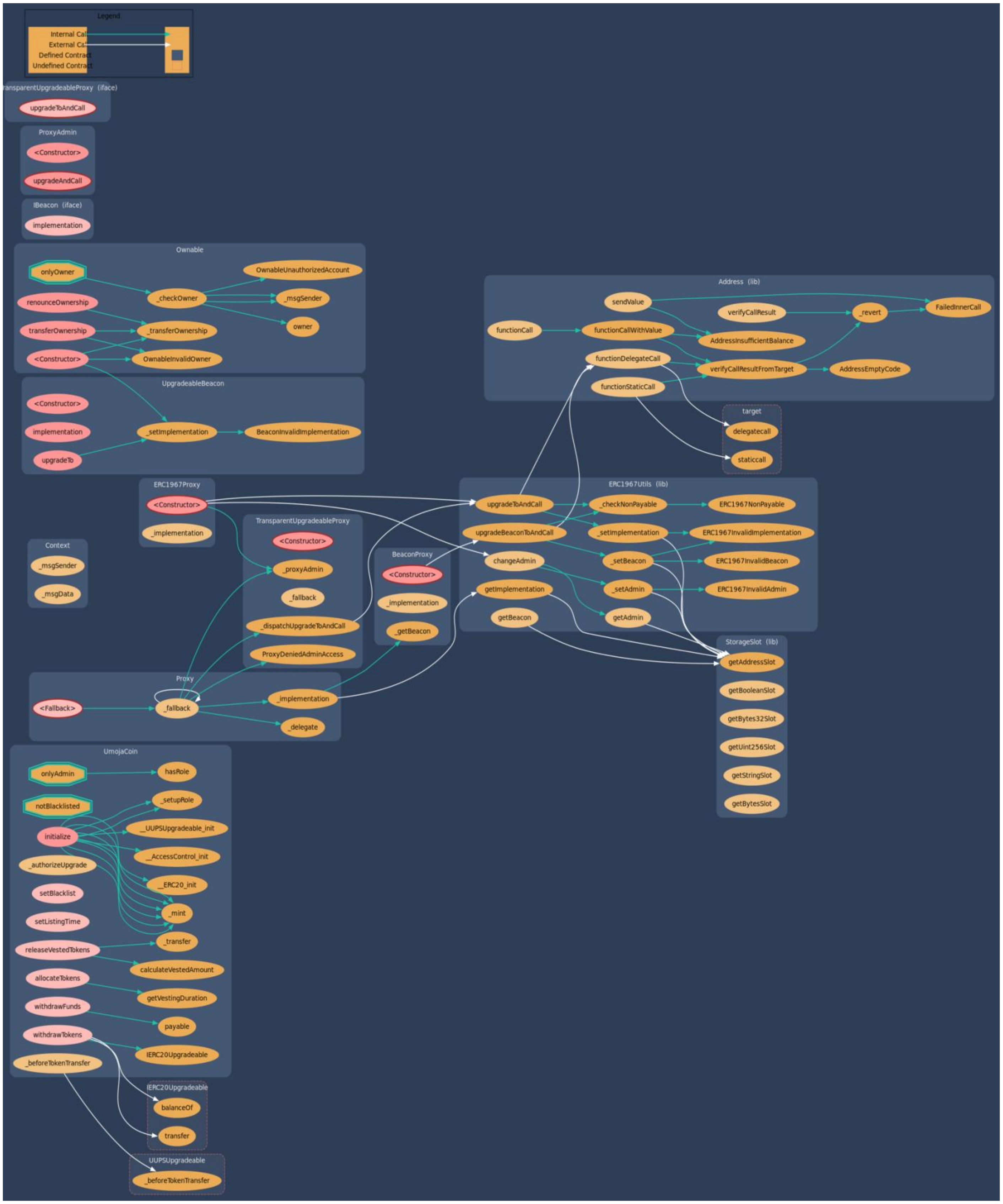
- ✓ **Constant/View/Pure functions:** Potentially constant/view/pure functions
- ✓ **Similar variable names:** Variable names are too similar
- ✓ **No return:** Function with 'returns' not returning
- ✓ **Guard conditions:** Ensure appropriate use of require/assert
- ✓ **Result not used:** The result of an operation not used
- ✓ **String length:** Bytes length != String length
- ✓ **Delete from dynamic array:** 'delete' leaves a gap in array
- ✓ **Data truncated:** Division on int/uint values truncates the result

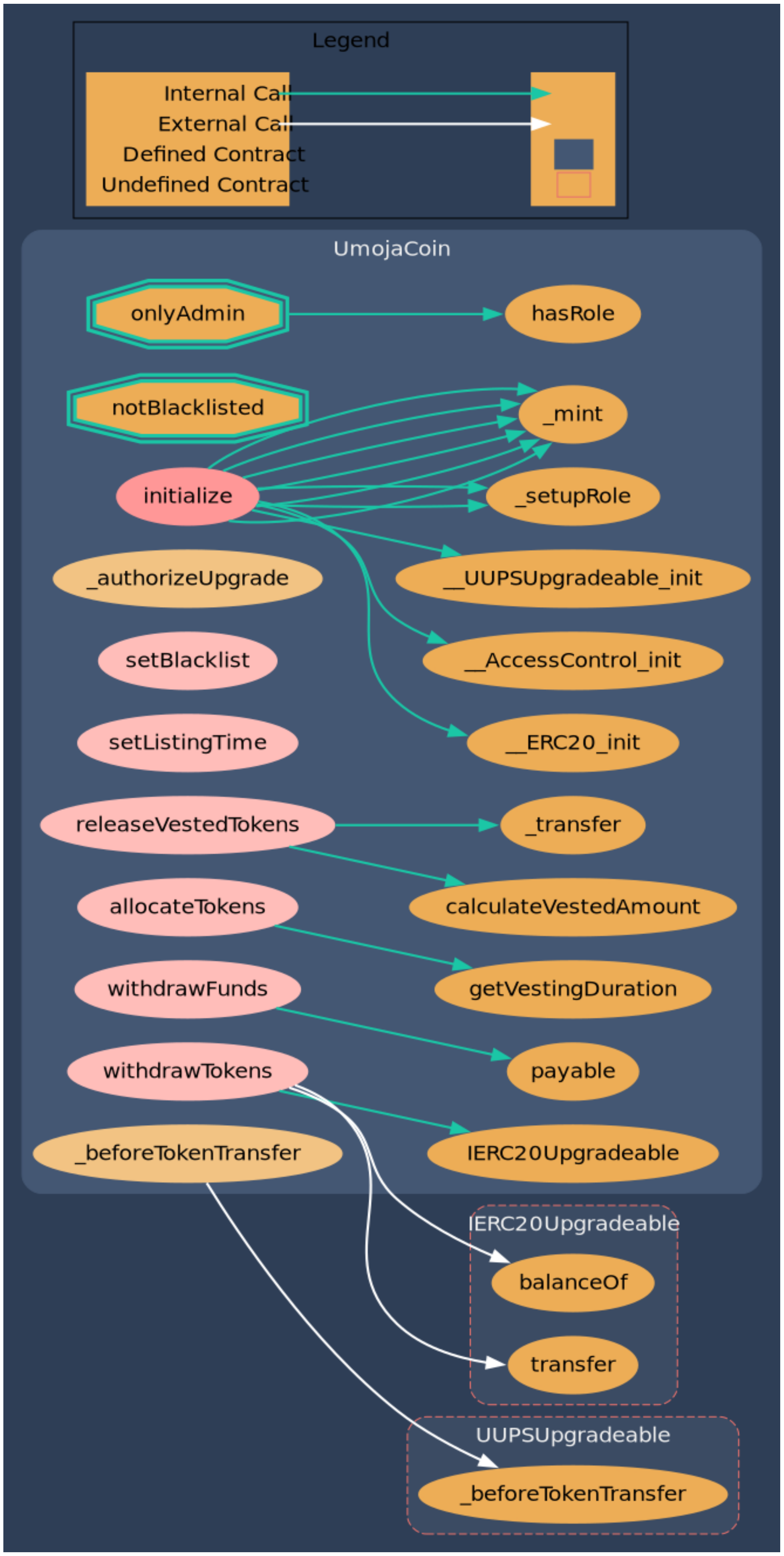
## 2- Inheritance graph



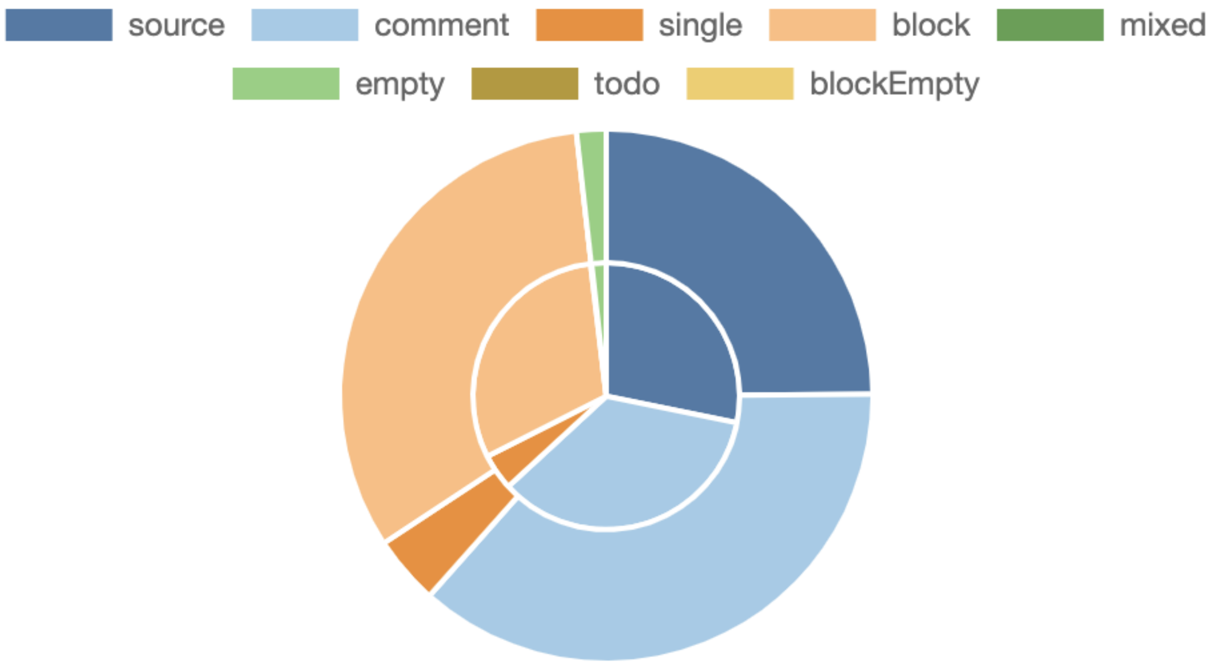


## 3- Call graph

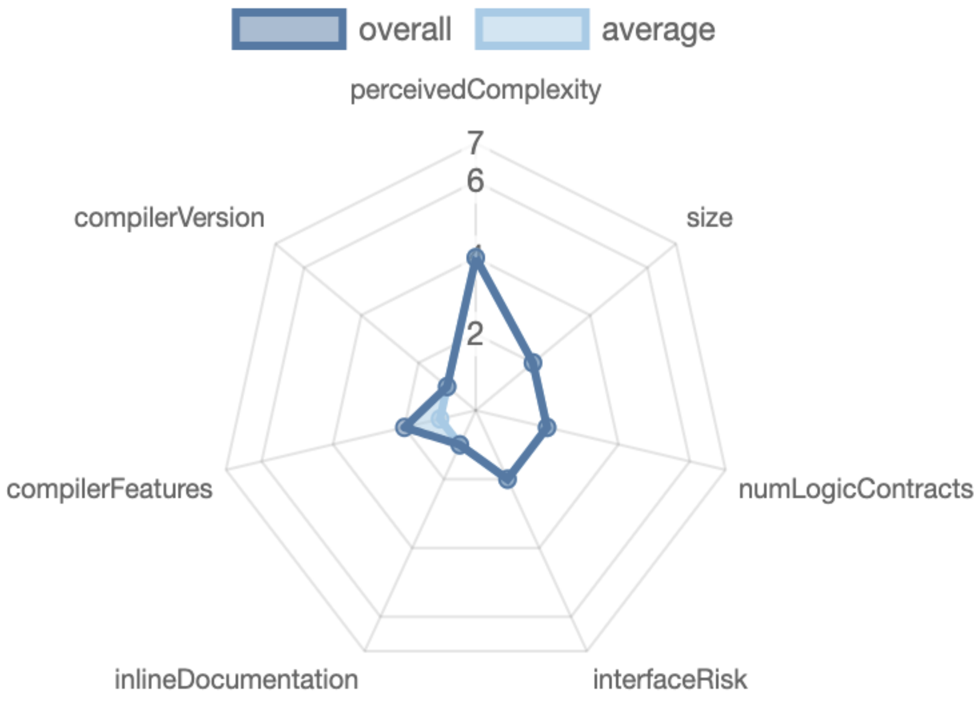

Legend
Internal Call
External Call
Defined Contract
Undefined Contract
TransparentUpgradeableProxy (iface)
upgradeToAndCall
ProxyAdmin
<Constructor>
upgradeAndCall
IBeacon (iface)
implementation
Ownable
onlyOwner
OwnableUnauthorizedAccount
renounceOwnership
_checkOwner
_msgSender
owner
transferOwnership
_transferOwnership
<Constructor>
OwnableInvalidOwner
UpgradeableBeacon
<Constructor>
implementation
_setImplementation
BeaconInvalidImplementation
upgradeTo
ERC1967Proxy
<Constructor>
_implementation
TransparentUpgradeableProxy
<Constructor>
_proxyAdmin
_fallback
_dispatchUpgradeToAndCall
ProxyDeniedAdminAccess
Context
_msgSender
_msgData
BeaconProxy
<Constructor>
_implementation
_getBeacon
Proxy
<Fallback>
_fallback
_implementation
_delegate
Address (lib)
sendValue
FailedInnerCall
verifyCallResult
_revert
functionCall
functionCallWithValue
AddressInsufficientBalance
functionDelegateCall
verifyCallResultFromTarget
AddressEmptyCode
functionStaticCall
target
delegatecall
staticcall
ERC1967Utils (lib)
upgradeToAndCall
_checkNonPayable
ERC1967NonPayable
upgradeBeaconToAndCall
_setImplementation
ERC1967InvalidImplementation
changeAdmin
_setBeacon
ERC1967InvalidBeacon
getImplementation
_setAdmin
ERC1967InvalidAdmin
getBeacon
getAdmin
StorageSlot (lib)
getAddressSlot
getBooleanSlot
getBytes32Slot
getUint256Slot
getStringSlot
getBytesSlot
UmojaCoin
onlyAdmin
hasRole
notBlacklisted
_setupRole
initialize
__UUPSUpgradeable_init
__AccessControl_init
_authorizeUpgrade
__ERC20_init
setBlacklist
_mint
setListingTime
_transfer
releaseVestedTokens
calculateVestedAmount
allocateTokens
getVestingDuration
withdrawFunds
payable
withdrawTokens
IERC20Upgradeable
_beforeTokenTransfer
IERC20Upgradeable
balanceOf
transfer
UUPSUpgradeable
_beforeTokenTransfer


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
UmojaCoin
onlyAdmin
hasRole
notBlacklisted
_mint
initialize
_setupRole
_authorizeUpgrade
__UUPSUpgradeable_init
setBlacklist
__AccessControl_init
setListingTime
__ERC20_init
releaseVestedTokens
_transfer
calculateVestedAmount
allocateTokens
getVestingDuration
withdrawFunds
payable
withdrawTokens
_beforeTokenTransfer
IERC20Upgradeable
IERC20Upgradeable
balanceOf
transfer
UUPSUpgradeable
_beforeTokenTransfer

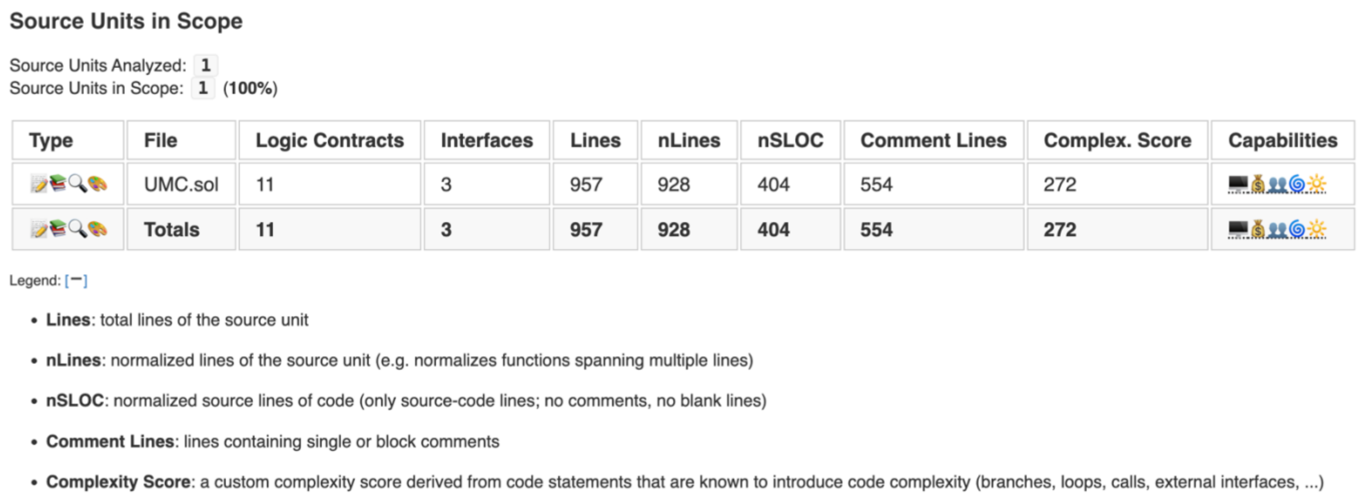
## Source lines



## Risk level



## Source units in scope



### Source Units in Scope

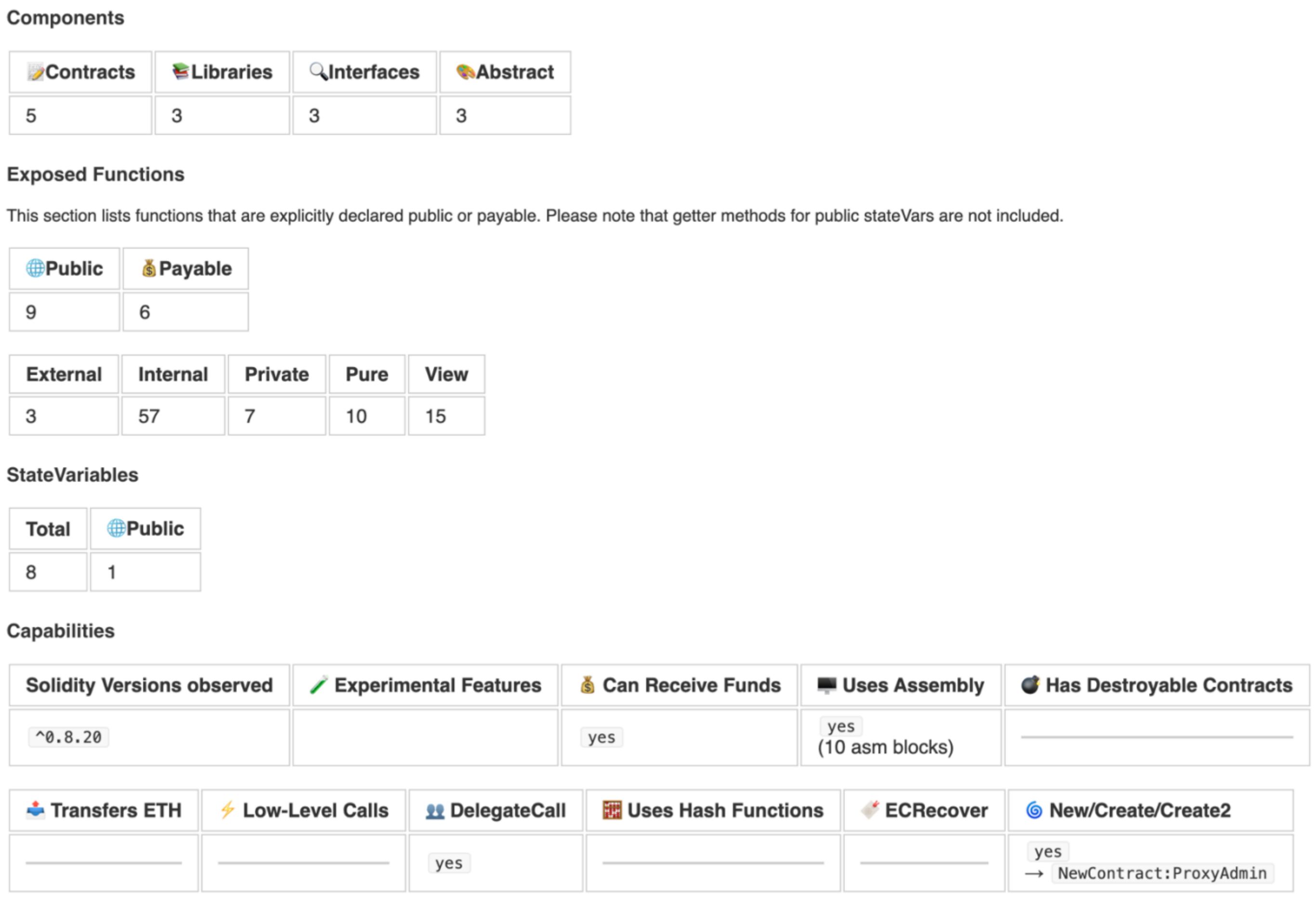
Source Units Analyzed: 1
Source Units in Scope: 1 (100%)

| Type | File | Logic Contracts | Interfaces | Lines | nLines | nSLOC | Comment Lines | Complex. Score | Capabilities |
|---|---|---|---|---|---|---|---|---|---|
| 📝📚🔍🎨 | UMC.sol | 11 | 3 | 957 | 928 | 404 | 554 | 272 | 🖥💰👥🌀🔆 |
| 📝📚🔍🎨 | **Totals** | **11** | **3** | **957** | **928** | **404** | **554** | **272** | 🖥💰👥🌀🔆 |

Legend: [−]

- **Lines**: total lines of the source unit
- **nLines**: normalized lines of the source unit (e.g. normalizes functions spanning multiple lines)
- **nSLOC**: normalized source lines of code (only source-code lines; no comments, no blank lines)
- **Comment Lines**: lines containing single or block comments
- **Complexity Score**: a custom complexity score derived from code statements that are known to introduce code complexity (branches, loops, calls, external interfaces, ...)

## Capabilities



### Components

| 📝Contracts | 📚Libraries | 🔍Interfaces | 🎨Abstract |
|---|---|---|---|
| 5 | 3 | 3 | 3 |

### Exposed Functions

This section lists functions that are explicitly declared public or payable. Please note that getter methods for public stateVars are not included.

| 🌐Public | 💰Payable |
|---|---|
| 9 | 6 |

| External | Internal | Private | Pure | View |
|---|---|---|---|---|
| 3 | 57 | 7 | 10 | 15 |

### StateVariables

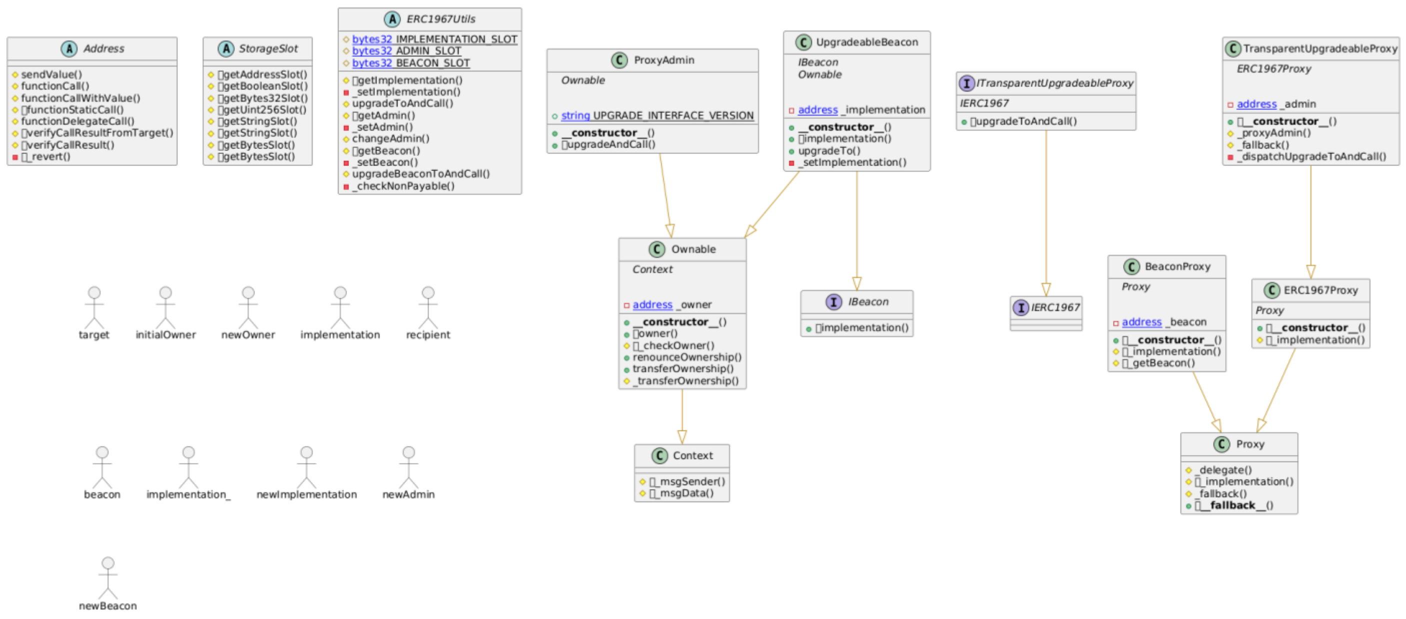
| Total | 🌐Public |
|---|---|
| 8 | 1 |

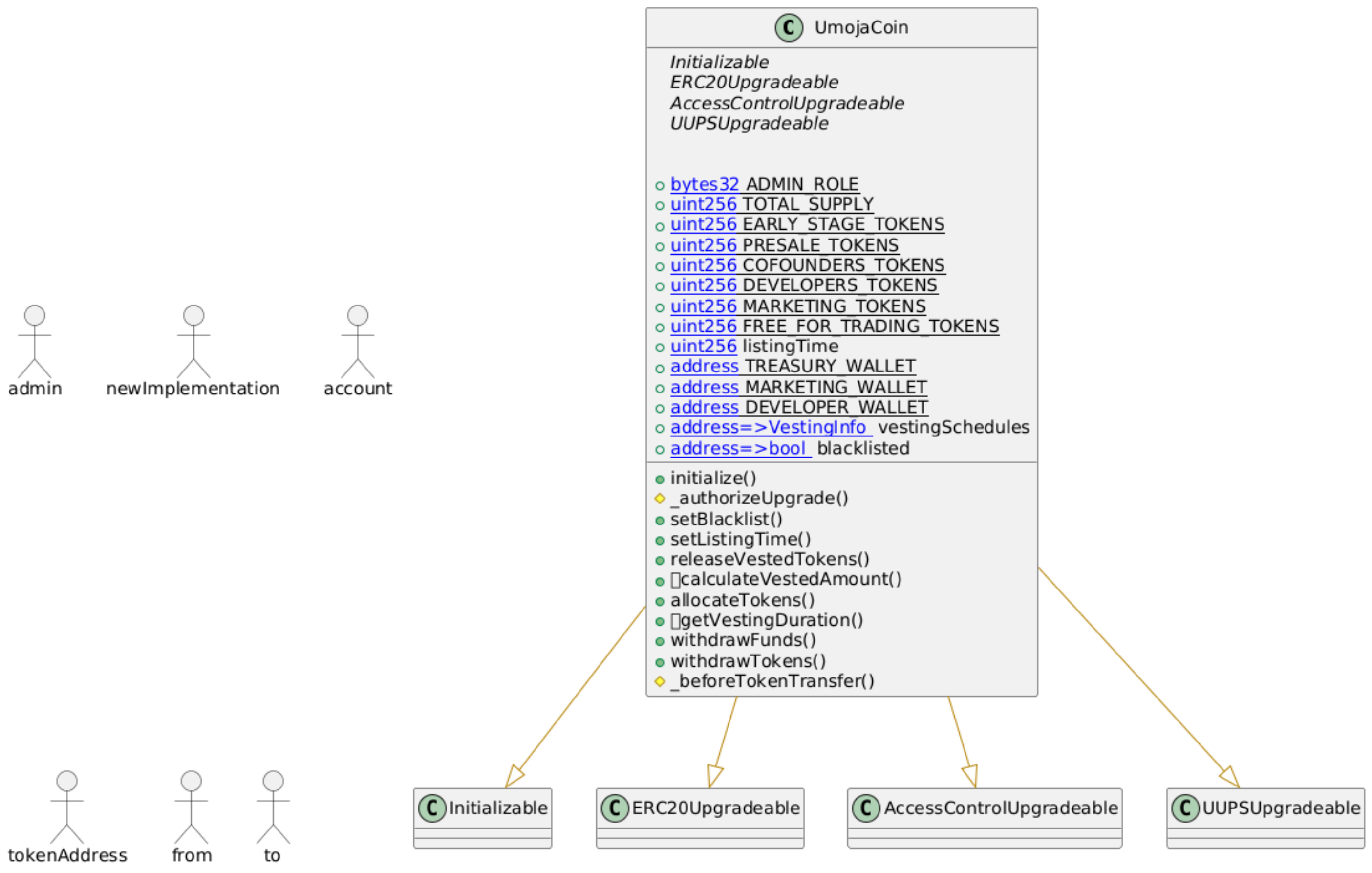
### Capabilities

| Solidity Versions observed | 🧪 Experimental Features | 💰 Can Receive Funds | 🖥 Uses Assembly | 💣 Has Destroyable Contracts |
|---|---|---|---|---|
| `^0.8.20` | | `yes` | `yes` (10 asm blocks) | |

| 📤 Transfers ETH | ⚡ Low-Level Calls | 👥 DelegateCall | 🧮 Uses Hash Functions | 🔖 ECRecover | 🌀 New/Create/Create2 |
|---|---|---|---|---|---|
| | | `yes` | | | `yes` → `NewContract:ProxyAdmin` |

# Unified Modeling Language (UML)





## Functions signature

| Function Name | Sighash | Function Signature |
| ------------- | ---------- | ------------------ |
| initialize | c4d66de8 | initialize(address) |
| setBlacklist | 153b0d1e | setBlacklist(address,bool) |
| setListingTime | e131d735 | setListingTime(uint256) |
| releaseVestedTokens | 54dd1da4 | releaseVestedTokens() |
| calculateVestedAmount | ba114be6 | calculateVestedAmount(uint256,uint256,uint256,uint8) |
| allocateTokens | 61738c87 | allocateTokens(address,uint256,uint8) |
| getVestingDuration | fe3aa8ea | getVestingDuration(uint8) |
| withdrawFunds | 155dd5ee | withdrawFunds(uint256) |
| withdrawTokens | 06b091f9 | withdrawTokens(address,uint256) |
| owner | 8da5cb5b | owner() |
| renounceOwnership | 715018a6 | renounceOwnership() |
| transferOwnership | f2fde38b | transferOwnership(address) |
| implementation | 5c60da1b | implementation() |
| implementation | 5c60da1b | implementation() |
| upgradeTo | 3659cfe6 | upgradeTo(address) |
| upgradeAndCall | 9623609d | upgradeAndCall(address,address,bytes) |
| upgradeToAndCall | 9df4d08e | upgradeToAndCall(bytes) |
| initialize | c4d66de8 | initialize(address) |
| setBlacklist | 153b0d1e | setBlacklist(address,bool) |
| setListingTime | e131d735 | setListingTime(uint256) |
| releaseVestedTokens | 54dd1da4 | releaseVestedTokens() |
| calculateVestedAmount | ba114be6 | calculateVestedAmount(uint256,uint256,uint256,uint8) |
| allocateTokens | 61738c87 | allocateTokens(address,uint256,uint8) |
| getVestingDuration | fe3aa8ea | getVestingDuration(uint8) |
| withdrawFunds | 155dd5ee | withdrawFunds(uint256) |
| withdrawTokens | 06b091f9 | withdrawTokens(address,uint256) |
| owner | 8da5cb5b | owner() |
| renounceOwnership | 715018a6 | renounceOwnership() |
| transferOwnership | f2fde38b | transferOwnership(address) |
| implementation | 5c60da1b | implementation() |
| implementation | 5c60da1b | implementation() |
| upgradeTo | 3659cfe6 | upgradeTo(address) |
| upgradeAndCall | 9623609d | upgradeAndCall(address,address,bytes) |
| upgradeToAndCall | 9df4d08e | upgradeToAndCall(bytes) |

## Automatic general report

```
Files Description Table

|  File Name  |  SHA-1 Hash  |
|-------------|--------------|
| /Users/macbook/Desktop/smart contracts/UMC.sol |
5d2ba2062d2d6d4d8306bedb0996931195154ec1 |
 Contracts Description Table

|  Contract  |         Type        |       Bases      |
|                  |
|:----------:|:-------------------:|:----------------:|:----------------
:|:----------------:|
|     └      |  **Function Name**  |  **Visibility**  |  **Mutability**
|  **Modifiers**  |
||||||
| **Address** | Library |  |||
| └ | sendValue | Internal 🔒 | 🛑  | |
| └ | functionCall | Internal 🔒 | 🛑  | |
| └ | functionCallWithValue | Internal 🔒 | 🛑  | |
| └ | functionStaticCall | Internal 🔒 |   | |
| └ | functionDelegateCall | Internal 🔒 | 🛑  | |
| └ | verifyCallResultFromTarget | Internal 🔒 |   | |
| └ | verifyCallResult | Internal 🔒 |   | |
| └ | _revert | Private 🔐 |   | |
||||||
| **Context** | Implementation |  |||
| └ | _msgSender | Internal 🔒 |   | |
| └ | _msgData | Internal 🔒 |   | |
||||||
| **StorageSlot** | Library |  |||
| └ | getAddressSlot | Internal 🔒 |   | |
| └ | getBooleanSlot | Internal 🔒 |   | |
| └ | getBytes32Slot | Internal 🔒 |   | |
| └ | getUint256Slot | Internal 🔒 |   | |
| └ | getStringSlot | Internal 🔒 |   | |
| └ | getStringSlot | Internal 🔒 |   | |
| └ | getBytesSlot | Internal 🔒 |   | |
| └ | getBytesSlot | Internal 🔒 |   | |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗ | 🛑  |NO❗ |
| └ | owner | Public ❗ |   |NO❗ |
| └ | _checkOwner | Internal 🔒 |   | |
| └ | renounceOwnership | Public ❗ | 🛑  | onlyOwner |
| └ | transferOwnership | Public ❗ | 🛑  | onlyOwner |
| └ | _transferOwnership | Internal 🔒 | 🛑  | |
||||||
| **IERC1967** | Interface |  |||
||||||
| **Proxy** | Implementation |  |||
| └ | _delegate | Internal 🔒 | 🛑  | |
| └ | _implementation | Internal 🔒 |   | |
| └ | _fallback | Internal 🔒 | 🛑  | |
| └ | <Fallback> | External ❗️ |  💵 |NO❗️ |
||||||
| **BeaconProxy** | Implementation | Proxy |||
| └ | <Constructor> | Public ❗️ |  💵 |NO❗️ |
| └ | _implementation | Internal 🔒 |   | |
| └ | _getBeacon | Internal 🔒 |   | |
||||||
| **IBeacon** | Interface |  |||
| └ | implementation | External ❗️ |   |NO❗️ |
||||||
| **UpgradeableBeacon** | Implementation | IBeacon, Ownable |||
| └ | <Constructor> | Public ❗️ | 🛑  | Ownable |
| └ | implementation | Public ❗️ |   |NO❗️ |
| └ | upgradeTo | Public ❗️ | 🛑  | onlyOwner |
| └ | _setImplementation | Private 🔐 | 🛑  | |
||||||
| **ERC1967Proxy** | Implementation | Proxy |||
| └ | <Constructor> | Public ❗️ |  💵 |NO❗️ |
| └ | _implementation | Internal 🔒 |   | |
||||||
| **ERC1967Utils** | Library |  |||
| └ | getImplementation | Internal 🔒 |   | |
| └ | _setImplementation | Private 🔐 | 🛑  | |
| └ | upgradeToAndCall | Internal 🔒 | 🛑  | |
| └ | getAdmin | Internal 🔒 |   | |
| └ | _setAdmin | Private 🔐 | 🛑  | |
| └ | changeAdmin | Internal 🔒 | 🛑  | |
| └ | getBeacon | Internal 🔒 |   | |
| └ | _setBeacon | Private 🔐 | 🛑  | |
| └ | upgradeBeaconToAndCall | Internal 🔒 | 🛑  | |
| └ | _checkNonPayable | Private 🔐 | 🛑  | |
||||||
| **ProxyAdmin** | Implementation | Ownable |||
| └ | <Constructor> | Public ❗️ | 🛑  | Ownable |
| └ | upgradeAndCall | Public ❗️ |  💵 | onlyOwner |
||||||
| **ITransparentUpgradeableProxy** | Interface | IERC1967 |||
| └ | upgradeToAndCall | External ❗️ |  💵 |NO❗️ |
||||||
| **TransparentUpgradeableProxy** | Implementation | ERC1967Proxy |||
| └ | <Constructor> | Public ❗️ |  💵 | ERC1967Proxy |
| └ | _proxyAdmin | Internal 🔒 | 🛑  | |
| └ | _fallback | Internal 🔒 | 🛑  | |
| └ | _dispatchUpgradeToAndCall | Private 🔐 | 🛑  | |
||||||
| **UmojaCoin** | Implementation | Initializable, ERC20Upgradeable,
AccessControlUpgradeable, UUPSUpgradeable |||
| └ | initialize | Public ❗️ | 🛑  | initializer |
| └ | _authorizeUpgrade | Internal 🔒 | 🛑  | onlyAdmin |
| └ | setBlacklist | External ❗️ | 🛑  | onlyAdmin |
| └ | setListingTime | External ❗️ | 🛑  | onlyAdmin |
| └ | releaseVestedTokens | External ❗️ | 🛑  | notBlacklisted |
| └ | calculateVestedAmount | Public ❗️ |   |NO❗️ |
| └ | allocateTokens | External ❗️ | 🛑  | onlyAdmin |
| └ | getVestingDuration | Public ❗️ |   |NO❗️ |
| └ | withdrawFunds | External ❗️ | 🛑  | onlyAdmin |
| └ | withdrawTokens | External ❗️ | 🛑  | onlyAdmin |
| └ | _beforeTokenTransfer | Internal 🔒 | 🛑  | |

 Legend

|  Symbol  |  Meaning  |
|:--------:|-----------|
|    🛑    | Function can modify state |
|    💵    | Function is payable |
```

## Conclusion

The contracts are written systematically. Team found no critical issues. So, it is good to go for production.

Since possible test cases can be unlimited and developer level documentation (code flow diagram with function level description) not provided, for such an extensive smart contract protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan Everything.

Security state of the reviewed contract is “Well Secured”.

✓ No volatile code.
✓ No high severity issues were found.

## Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as of the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against the team on the basis of what it says or doesn’t say, or how team produced it, and it is important for you to conduct your own independent investigations before making any decisions. team go into more detail on this in the below disclaimer below – please make sure to read it in full.

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Saferico and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Saferico s) owe no duty of care towards you or any other person, nor does Saferico make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Saferico hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Saferico hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Saferico, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.