



Cyberscope

# Audit Report

# **Battle World**

August 2022

Audited by © cyberscope

# Table of Contents

|                                    |          |
|------------------------------------|----------|
| <b>Table of Contents</b>           | <b>1</b> |
| <b>Introduction</b>                | <b>2</b> |
| <b>Recommendation Across Files</b> | <b>2</b> |
| <b>Corresponding Audits</b>        | <b>3</b> |
| <b>Audit NFT1155</b>               | <b>3</b> |
| <b>Audit NFTDistributor</b>        | <b>3</b> |
| <b>Audit RaffleTicket</b>          | <b>3</b> |
| <b>Audit RaffleDistributor</b>     | <b>3</b> |
| <b>Audit RaffleRefund</b>          | <b>3</b> |
| <b>Audit Vault</b>                 | <b>3</b> |
| <b>Disclaimer</b>                  | <b>4</b> |
| <b>About Cyberscope</b>            | <b>5</b> |

# Introduction

Cyberscope audited six sections of the Battle World ecosystem. The Battle Word ecosystem provides functionality to purchase tickets and buy NFTs.

- The Vault contract is responsible for saving the accumulated funds.
- The NFT1155 contract is the NFT contract.
- The RaffleTicket represents the digital currency for the ecosystem.
- The RaffleDistributor contract is responsible for the ticket purchases mechanism.
- The RaffleRefund contract is responsible to redeem tickets and return the corresponding funds to the user.
- The NFTDistributor is responsible for selling NFTs using raffle tickets as the currency.

## Recommendation Across Files

The owners and the permissions functionality is repetitive across all the contracts. It could be moved to a separate contract so that all the contracts will share exactly the sale functionality.

# Corresponding Audits

## Audit NFT1155

<https://github.com/cyberscope-io/audits/tree/main/bwo/nft1155.pdf>

## Audit NFTDistributor

<https://github.com/cyberscope-io/audits/tree/main/bwo/nftDistributor.pdf>

## Audit RaffleTicket

<https://github.com/cyberscope-io/audits/tree/main/bwo/raffleTicket.pdf>

## Audit RaffleDistributor

<https://github.com/cyberscope-io/audits/tree/main/bwo/raffleDistributor.pdf>

## Audit RaffleRefund

<https://github.com/cyberscope-io/audits/tree/main/bwo/raffleRefund.pdf>

## Audit Vault

<https://github.com/cyberscope-io/audits/tree/main/bwo/valut.pdf>

## Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

# About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.



The Cyberscope team

<https://www.cyberscope.io>



Cyberscope

# Audit Report

## **NFT1155**

August 2022

SHA256    a71293c8be464c20a24c058fa451a834f000d47d42e15183410a60475a26a214

Audited by   © cyberscope

# Table of Contents

|                                                         |           |
|---------------------------------------------------------|-----------|
| <b>Table of Contents</b>                                | <b>1</b>  |
| <b>Contract Review</b>                                  | <b>2</b>  |
| <b>Audit Updates</b>                                    | <b>2</b>  |
| <b>Source Files</b>                                     | <b>3</b>  |
| <b>Introduction</b>                                     | <b>5</b>  |
| <b>Contract Roles</b>                                   | <b>6</b>  |
| <b>Contract Diagnostics</b>                             | <b>7</b>  |
| <b>L04 - Conformance to Solidity Naming Conventions</b> | <b>8</b>  |
| <b>Description</b>                                      | <b>8</b>  |
| <b>Recommendation</b>                                   | <b>8</b>  |
| <b>Contract Functions</b>                               | <b>9</b>  |
| <b>Contract Flow</b>                                    | <b>13</b> |
| <b>Summary</b>                                          | <b>14</b> |
| <b>Disclaimer</b>                                       | <b>15</b> |
| <b>About Cyberscope</b>                                 | <b>16</b> |

## Contract Review

|                       |                                                                                                                                                                                       |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Contract Name</b>  | NFT1155                                                                                                                                                                               |
| <b>Testing Deploy</b> | <a href="https://testnet.bscscan.com/address/0xE339b84254bC74f2F2733D62BeCcBc996f6f0D3A#code">https://testnet.bscscan.com/address/0xE339b84254bC74f2F2733D62BeCcBc996f6f0D3A#code</a> |
| <b>Symbol</b>         | TS                                                                                                                                                                                    |
| <b>Domain</b>         | <a href="https://battleworld.game">https://battleworld.game</a>                                                                                                                       |

## Audit Updates

|                      |                 |
|----------------------|-----------------|
| <b>Initial Audit</b> | 4th August 2022 |
| <b>Corrected</b>     |                 |

## Source Files

| Filename                                                                 | SHA256                                                           |
|--------------------------------------------------------------------------|------------------------------------------------------------------|
| @openzeppelin/contracts/interfaces/IERC2981.sol                          | a8d1ed2cf4ae5718b5c1e8e148e7f5817e934e8d933729eb5c3ac389046041b1 |
| @openzeppelin/contracts/token/common/ERC2981.sol                         | 56fbc83102cc78244419c4c9ab9ee48e53874bb9fa11eda45c85c1c5cf87c076 |
| @openzeppelin/contracts/token/ERC1155/ERC1155.sol                        | 3a7b1481259da24728a0bac33ac9728c0faf71d436e4f198209815f732240a24 |
| @openzeppelin/contracts/token/ERC1155/extensions/IERC1155MetadataURI.sol | 6987fbfa647d3da51e8c270371ac48c5fcd26fb046cf54644b39aa098ae30324 |
| @openzeppelin/contracts/token/ERC1155/IERC1155.sol                       | fd6a1801f1f2f8af0a3ece0b254da06ec24568aec02cfe94827061379aebc6f3 |
| @openzeppelin/contracts/token/ERC1155/IERC1155Receiver.sol               | 578834a1bcdac6a22de5e07ae63bbbd4d41615f35950afc6e6c068d92619b334 |
| @openzeppelin/contracts/utils/Address.sol                                | 1e0922f6c0bf6b1b8b4d480dcabb691b1359195a297bde6dc5172e79f3a1f826 |
| @openzeppelin/contracts/utils/Context.sol                                | 1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a |
| @openzeppelin/contracts/utils/introspection                              | 8806a632d7b656cadb8133ff8f2acae4405b3a64d8709d93b0fa6a216a8a6154 |

|                                                                |                                                                  |
|----------------------------------------------------------------|------------------------------------------------------------------|
| <b>ection/ERC165.sol</b>                                       |                                                                  |
| <b>@openzeppelin/contracts/utils/introspection/IERC165.sol</b> | 701e025d13ec6be09ae892eb029cd83b3064325801d73654847a5fb11c58b1e5 |
| <b>contracts/NFT1155.sol</b>                                   | a71293c8be464c20a24c058fa451a834f000d47d42e15183410a60475a26a214 |

# Introduction

The NFT1155 contract is an nft contract. The contract owner has the ability to authorize addresses to have Mint access. In addition the contract owner has the ability to approve marketplaces addresses in order to sell Nfts. The value of Royalty Beneficiary is defined by the contract owners during the contract deployment and it can be changed by the owner.

# Contract Roles

Role owner:

- The contract owners can pause the marketplace approval and the Nft transfer functionality.
- The contract owners can authorize mintAccess.

Role mintAccess:

- The authorized address can mint Nfts.

# Contract Diagnostics

● Critical    ● Medium    ● Minor

| Severity | Code | Description                                |
|----------|------|--------------------------------------------|
| ●        | L04  | Conformance to Solidity Naming Conventions |

## L04 - Conformance to Solidity Naming Conventions

**Criticality**

minor

**Location**

contracts/NFT1155.sol#L34,296,289,37,266,36

### Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow `_` at the beginning of the `mixed_case` match for private variables and unused parameters.

```
GaslessListing
_i
ApprovedMP
_operator
_id
_owner
MintAccess
```

### Recommendation

Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>.

# Contract Functions

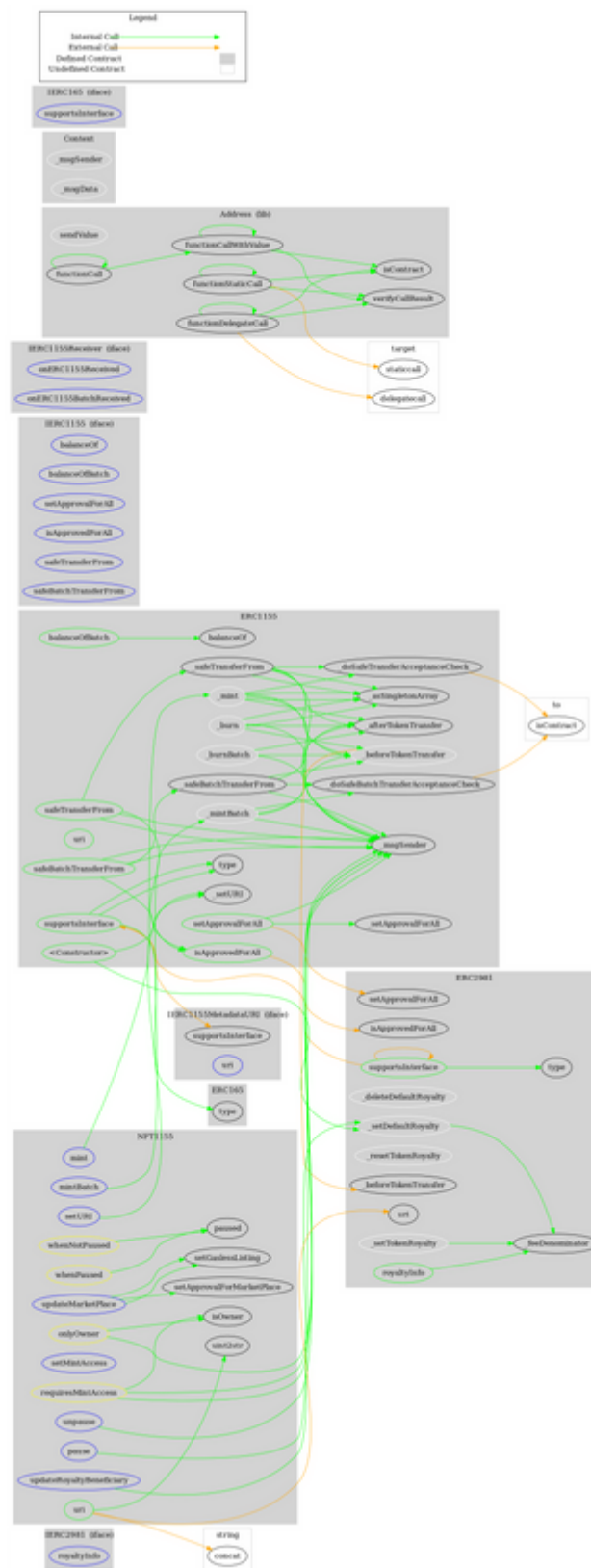
| Contract        | Type                   | Bases                                                       |            |           |
|-----------------|------------------------|-------------------------------------------------------------|------------|-----------|
|                 | Function Name          | Visibility                                                  | Mutability | Modifiers |
|                 |                        |                                                             |            |           |
| <b>IERC2981</b> | Interface              | IERC165                                                     |            |           |
|                 | royaltyInfo            | External                                                    |            | -         |
|                 |                        |                                                             |            |           |
| <b>ERC2981</b>  | Implementation         | IERC2981,<br>ERC165                                         |            |           |
|                 | supportsInterface      | Public                                                      |            | -         |
|                 | royaltyInfo            | Public                                                      |            | -         |
|                 | _feeDenominator        | Internal                                                    |            |           |
|                 | _setDefaultRoyalty     | Internal                                                    | ✓          |           |
|                 | _deleteDefaultRoyalty  | Internal                                                    | ✓          |           |
|                 | _setTokenRoyalty       | Internal                                                    | ✓          |           |
|                 | _resetTokenRoyalty     | Internal                                                    | ✓          |           |
|                 |                        |                                                             |            |           |
| <b>ERC1155</b>  | Implementation         | Context,<br>ERC165,<br>IERC1155,<br>IERC1155M<br>etadataURI |            |           |
|                 | <Constructor>          | Public                                                      | ✓          | -         |
|                 | supportsInterface      | Public                                                      |            | -         |
|                 | uri                    | Public                                                      |            | -         |
|                 | balanceOf              | Public                                                      |            | -         |
|                 | balanceOfBatch         | Public                                                      |            | -         |
|                 | setApprovalForAll      | Public                                                      | ✓          | -         |
|                 | isApprovedForAll       | Public                                                      |            | -         |
|                 | safeTransferFrom       | Public                                                      | ✓          | -         |
|                 | safeBatchTransferFrom  | Public                                                      | ✓          | -         |
|                 | _safeTransferFrom      | Internal                                                    | ✓          |           |
|                 | _safeBatchTransferFrom | Internal                                                    | ✓          |           |
|                 | _setURI                | Internal                                                    | ✓          |           |

|                         |                                     |          |   |   |
|-------------------------|-------------------------------------|----------|---|---|
|                         | _mint                               | Internal | ✓ |   |
|                         | _mintBatch                          | Internal | ✓ |   |
|                         | _burn                               | Internal | ✓ |   |
|                         | _burnBatch                          | Internal | ✓ |   |
|                         | _setApprovalForAll                  | Internal | ✓ |   |
|                         | _beforeTokenTransfer                | Internal | ✓ |   |
|                         | _afterTokenTransfer                 | Internal | ✓ |   |
|                         | _doSafeTransferAcceptanceCheck      | Private  | ✓ |   |
|                         | _doSafeBatchTransferAcceptanceCheck | Private  | ✓ |   |
|                         | _asSingletonArray                   | Private  |   |   |
|                         |                                     |          |   |   |
| <b>IERC1155Meta</b>     | Interface                           | IERC1155 |   |   |
| <b>dataURI</b>          | uri                                 | External |   | - |
|                         |                                     |          |   |   |
| <b>IERC1155</b>         | Interface                           | IERC165  |   |   |
|                         | balanceOf                           | External |   | - |
|                         | balanceOfBatch                      | External |   | - |
|                         | setApprovalForAll                   | External | ✓ | - |
|                         | isApprovedForAll                    | External |   | - |
|                         | safeTransferFrom                    | External | ✓ | - |
|                         | safeBatchTransferFrom               | External | ✓ | - |
|                         |                                     |          |   |   |
| <b>IERC1155Receiver</b> | Interface                           | IERC165  |   |   |
|                         | onERC1155Received                   | External | ✓ | - |
|                         | onERC1155BatchReceived              | External | ✓ | - |
|                         |                                     |          |   |   |
| <b>Address</b>          | Library                             |          |   |   |
|                         | isContract                          | Internal |   |   |
|                         | sendValue                           | Internal | ✓ |   |
|                         | functionCall                        | Internal | ✓ |   |
|                         | functionCall                        | Internal | ✓ |   |
|                         | functionCallWithValue               | Internal | ✓ |   |
|                         | functionCallWithValue               | Internal | ✓ |   |

|                |                           |                     |   |               |
|----------------|---------------------------|---------------------|---|---------------|
|                | functionStaticCall        | Internal            |   |               |
|                | functionStaticCall        | Internal            |   |               |
|                | functionDelegateCall      | Internal            | ✓ |               |
|                | functionDelegateCall      | Internal            | ✓ |               |
|                | verifyCallResult          | Internal            |   |               |
|                |                           |                     |   |               |
| <b>Context</b> | Implementation            |                     |   |               |
|                | _msgSender                | Internal            |   |               |
|                | _msgData                  | Internal            |   |               |
|                |                           |                     |   |               |
| <b>ERC165</b>  | Implementation            | IERC165             |   |               |
|                | supportsInterface         | Public              |   | -             |
|                |                           |                     |   |               |
| <b>IERC165</b> | Interface                 |                     |   |               |
|                | supportsInterface         | External            |   | -             |
|                |                           |                     |   |               |
| <b>NFT1155</b> | Implementation            | ERC1155,<br>ERC2981 |   |               |
|                | <Constructor>             | Public              | ✓ | ERC1155       |
|                | supportsInterface         | Public              |   | -             |
|                | updateRoyaltyBeneficiary  | External            | ✓ | onlyOwner     |
|                | isOwner                   | Public              |   | -             |
|                | paused                    | Public              |   | -             |
|                | pause                     | External            | ✓ | onlyOwner     |
|                | unpause                   | External            | ✓ | onlyOwner     |
|                | setMintAccess             | External            | ✓ | onlyOwner     |
|                | setGaslessListing         | Internal            | ✓ |               |
|                | setApprovalForMarketPlace | Internal            | ✓ |               |
|                | updateMarketPlace         | External            | ✓ | onlyOwner     |
|                | setURI                    | External            | ✓ | onlyOwner     |
|                | _beforeTokenTransfer      | Internal            | ✓ | whenNotPaused |
|                | uint2str                  | Internal            |   |               |
|                | uri                       | Public              |   | -             |
|                | isApprovedForAll          | Public              |   | -             |
|                | setApprovalForAll         | Public              | ✓ | whenNotPaused |

|  |           |          |   |                    |
|--|-----------|----------|---|--------------------|
|  |           |          |   | ed                 |
|  | mint      | External | ✓ | requiresMintAccess |
|  | mintBatch | External | ✓ | requiresMintAccess |

# Contract Flow



## Summary

The NFT115 contract implements a nft contract. It provides functionality to mint Nfts for the allowed addresses. The audit investigates the main features, mentions security recommendation, performance improvements and potential optimizations.

# Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

# About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.



The Cyberscope team

<https://www.cyberscope.io>



Cyberscope

# Audit Report

## **NFTDistributor**

August 2022

SHA256    b848192e9e5a1ac25fa886c82f298d2d727751e34512f2dcd212ef8da8b0818f

Audited by   © cyberscope

# Table of Contents

|                                                         |           |
|---------------------------------------------------------|-----------|
| <b>Table of Contents</b>                                | <b>1</b>  |
| <b>Contract Review</b>                                  | <b>2</b>  |
| <b>Source Files</b>                                     | <b>2</b>  |
| <b>Audit Updates</b>                                    | <b>2</b>  |
| <b>Introduction</b>                                     | <b>3</b>  |
| <b>Request Verification</b>                             | <b>4</b>  |
| <b>Contract Roles</b>                                   | <b>5</b>  |
| <b>Contract Diagnostics</b>                             | <b>6</b>  |
| <b>USB - User Sufficient Balance</b>                    | <b>7</b>  |
| <b>Description</b>                                      | <b>7</b>  |
| <b>Recommendation</b>                                   | <b>7</b>  |
| <b>L04 - Conformance to Solidity Naming Conventions</b> | <b>8</b>  |
| <b>Description</b>                                      | <b>8</b>  |
| <b>Recommendation</b>                                   | <b>8</b>  |
| <b>Contract Functions</b>                               | <b>9</b>  |
| <b>Contract Flow</b>                                    | <b>10</b> |
| <b>Summary</b>                                          | <b>11</b> |
| <b>Disclaimer</b>                                       | <b>12</b> |
| <b>About Cyberscope</b>                                 | <b>13</b> |

## Contract Review

|                      |                                                                                                                                                                             |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Contract Name</b> | NFTDistributor                                                                                                                                                              |
| <b>Test Deploy</b>   | <a href="https://testnet.bscscan.com/address/0x24D66Bd585510daCa4B1365da9d39E945ad413f9">https://testnet.bscscan.com/address/0x24D66Bd585510daCa4B1365da9d39E945ad413f9</a> |
| <b>Domain</b>        | <a href="https://battleworld.game">https://battleworld.game</a>                                                                                                             |

## Source Files

|                     |                                                                  |
|---------------------|------------------------------------------------------------------|
| <b>Filename</b>     | <b>SHA256</b>                                                    |
| <b>contract.sol</b> | b848192e9e5a1ac25fa886c82f298d2d727751e34512f2dcd212ef8da8b0818f |

## Audit Updates

|                      |                 |
|----------------------|-----------------|
| <b>Initial Audit</b> | 4th August 2022 |
| <b>Corrected</b>     |                 |

# Introduction

The NFTDistributor contract implements a NFT purchase mechanism. The users can deposit their tickets in order to receive NFTs. During the purchase process, the corresponding tickets are burned from the user's balance. The ratio between NFTs and tickets is defined off-chain. The in-chain method is responsible for burning the deposited amount and minting the NFTs. The users have to provide the token ids, the amounts of tokens and a message that must be verified in order to proceed with the transaction.

## Request Verification

The verification process is based on an off-chain configuration. The contract owners are responsible for updating the in-chain factor in order to validate correctly the provided message.

The verification algorithm is using the markle tree mechanism.

<https://github.com/protofire/zeppelin-solidity/blob/master/contracts/MerkleProof.sol>

According to the markle algorithm, the off-chain mechanism pre-defines all the `index, recipient, amount` combinations.

Hence, only pre-defined users have the ability to redeem tickets in specific amounts.

# Contract Roles

Role owner:

- The contract owners can pause the purchase mechanism.
- The contract owners can invalidate the validation factor and reset the saved address that mint Nfts.

# Contract Diagnostics

● Critical ● Medium ● Informative

| Severity | Code | Description                                |
|----------|------|--------------------------------------------|
| ●        | USB  | User Sufficient Balance                    |
| ●        | L04  | Conformance to Solidity Naming Conventions |

## USB - User Sufficient Balance

|                    |                   |
|--------------------|-------------------|
| <b>Criticality</b> | informative       |
| <b>Location</b>    | contract.sol#L333 |

### Description

The contract is based on the fact that the burnFrom method from the raffleTicket will revert in case that the user's balance is insufficient.

```
require(  
    IRaffleTicket(RAFFLE_TICKET_ADDRESS).burnFrom(  
        _msgSender(),  
        rafflecount  
    ),  
    "NFTDistributor#claimNFT: Burning Raffle Ticket Failed"  
);
```

### Recommendation

The contract could proactively check if the user's ticket balance is sufficient.

## L04 - Conformance to Solidity Naming Conventions

**Criticality**

minor

**Location**

contract.sol#L136,134,135

### Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow \_ at the beginning of the mixed\_case match for private variables and unused parameters.

```
NFT_ADDRESS  
RAFFLE_TICKET_ADDRESS  
ClaimMerkleRoot
```

### Recommendation

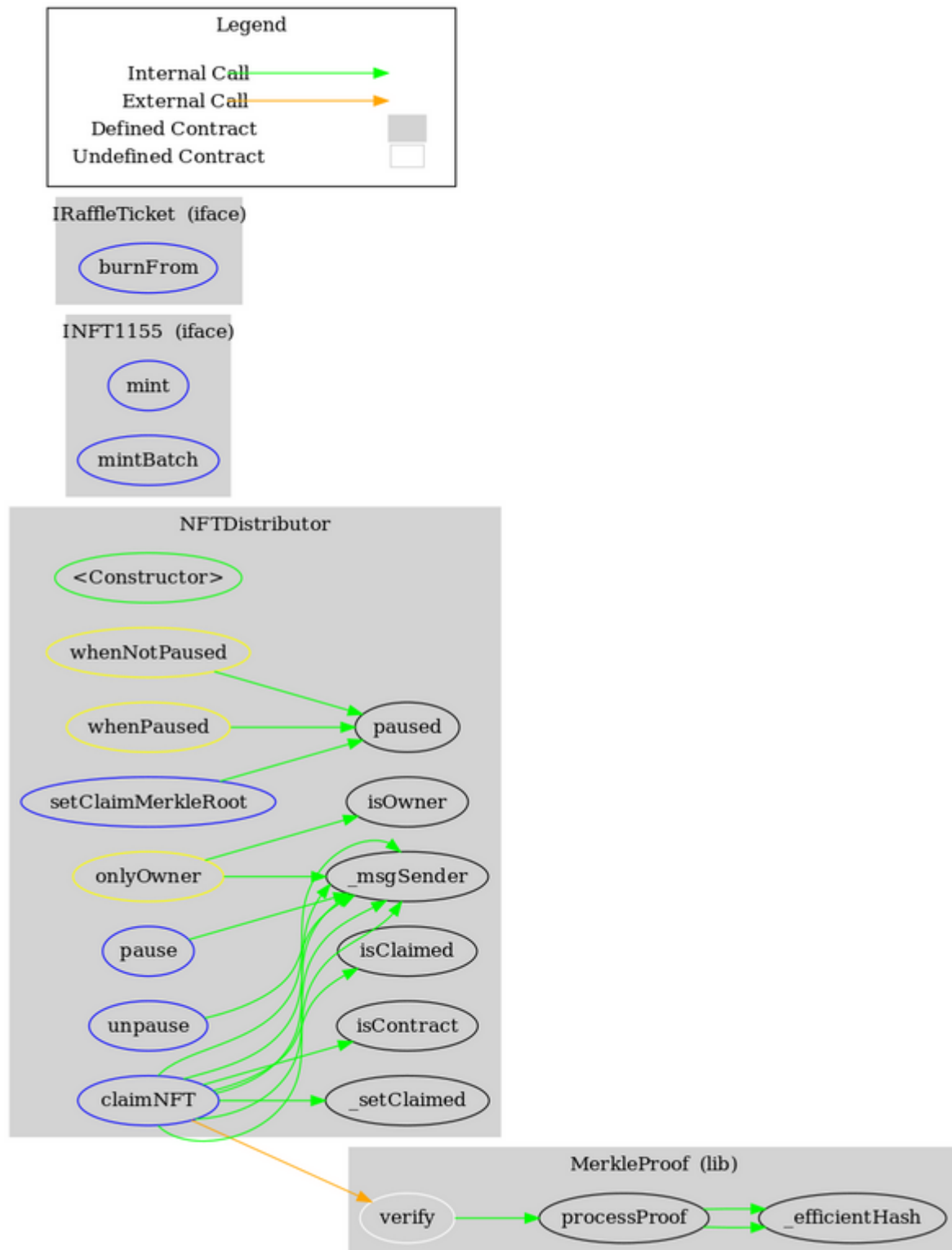
Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>.

# Contract Functions

| Contract              | Type               | Bases      |            |               |
|-----------------------|--------------------|------------|------------|---------------|
|                       | Function Name      | Visibility | Mutability | Modifiers     |
|                       |                    |            |            |               |
| <b>MerkleProof</b>    | Library            |            |            |               |
|                       | verify             | Internal   |            |               |
|                       | processProof       | Internal   |            |               |
|                       | _efficientHash     | Private    |            |               |
|                       |                    |            |            |               |
| <b>INFT1155</b>       | Interface          |            |            |               |
|                       | mint               | External   | ✓          | -             |
|                       | mintBatch          | External   | ✓          | -             |
|                       |                    |            |            |               |
| <b>IRaffleTicket</b>  | Interface          |            |            |               |
|                       | burnFrom           | External   | ✓          | -             |
|                       |                    |            |            |               |
| <b>NFTDistributor</b> | Implementation     |            |            |               |
|                       | <Constructor>      | Public     | ✓          | -             |
|                       | _msgSender         | Internal   |            |               |
|                       | isOwner            | Public     |            | -             |
|                       | paused             | Public     |            | -             |
|                       | pause              | External   | ✓          | onlyOwner     |
|                       | unpause            | External   | ✓          | onlyOwner     |
|                       | isContract         | Internal   |            |               |
|                       | setClaimMerkleRoot | External   | ✓          | onlyOwner     |
|                       | isClaimed          | Public     |            | -             |
|                       | _setClaimed        | Private    | ✓          |               |
|                       | claimNFT           | External   | ✓          | whenNotPaused |

# Contract Flow



## Summary

The Vault contract implements a purchase mechanism. It provides functionality to purchase Nfts by burning the corresponding tickets. The contract should thoroughly check balances before every transaction. The audit investigates the main features, mentions security recommendation, performance improvements and potential optimizations.

# Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

# About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.



The Cyberscope team

<https://www.cyberscope.io>



Cyberscope

Audit Report

# RaffleDistributor

August 2022

SHA256 50acaa2a55383eb1779841d1a37ef93a090ebd1602dd3ce9dc47d7f6cb39a6ea

Audited by © cyberscope

# Table of Contents

|                                                         |           |
|---------------------------------------------------------|-----------|
| <b>Table of Contents</b>                                | <b>1</b>  |
| <b>Contract Review</b>                                  | <b>2</b>  |
| <b>Source Files</b>                                     | <b>2</b>  |
| <b>Audit Updates</b>                                    | <b>2</b>  |
| <b>Introduction</b>                                     | <b>3</b>  |
| <b>Contract Roles</b>                                   | <b>4</b>  |
| <b>Contract Diagnostics</b>                             | <b>5</b>  |
| <b>L04 - Conformance to Solidity Naming Conventions</b> | <b>6</b>  |
| <b>Description</b>                                      | <b>6</b>  |
| <b>Recommendation</b>                                   | <b>6</b>  |
| <b>Contract Functions</b>                               | <b>7</b>  |
| <b>Contract Flow</b>                                    | <b>8</b>  |
| <b>Summary</b>                                          | <b>9</b>  |
| <b>Disclaimer</b>                                       | <b>10</b> |
| <b>About Cyberscope</b>                                 | <b>11</b> |

## Contract Review

|                       |                                                                                                                                                                             |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Contract Name</b>  | RaffleDistributor                                                                                                                                                           |
| <b>Testing Deploy</b> | <a href="https://testnet.bscscan.com/address/0x349fC037eB55B2e4b3810EC50723B79a5919fc60">https://testnet.bscscan.com/address/0x349fC037eB55B2e4b3810EC50723B79a5919fc60</a> |
| <b>Domain</b>         | <a href="https://battleworld.game">https://battleworld.game</a>                                                                                                             |

## Source Files

|                     |                                                                  |
|---------------------|------------------------------------------------------------------|
| <b>Filename</b>     | <b>SHA256</b>                                                    |
| <b>contract.sol</b> | 50acaa2a55383eb1779841d1a37ef93a090ebd1602dd3ce9dc47d7f6cb39a6ea |

## Audit Updates

|                      |                 |
|----------------------|-----------------|
| <b>Initial Audit</b> | 4th August 2022 |
| <b>Corrected</b>     |                 |

# Introduction

The RaffleDistributor contract implements a ticket purchase mechanism. The buyer provides the quantity of tickets and value of the tickets in order to receive tickets. The tickets are native tokens that are minted to the buyer's address.

The ticket price, ticket address, the vault address and maximum amount of tickets per user is defined once during the contract deployment.

The contract owners have the ability to pause the purchase mechanism.

# Contract Roles

Role owner:

- The contract owners can pause the purchase mechanism.

# Contract Diagnostics

● Critical   ● Medium   ● Minor

| Severity | Code | Description                                |
|----------|------|--------------------------------------------|
| ●        | L04  | Conformance to Solidity Naming Conventions |

## L04 - Conformance to Solidity Naming Conventions

**Criticality**

minor

**Location**

contract.sol#L35,32,36,33

### Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow \_ at the beginning of the mixed\_case match for private variables and unused parameters.

```
VAULT  
MAX_RAFFLE_PER_USER  
RAFFLE_TICKET_ADDRESS  
RAFFLE_TICKET_PRICE
```

### Recommendation

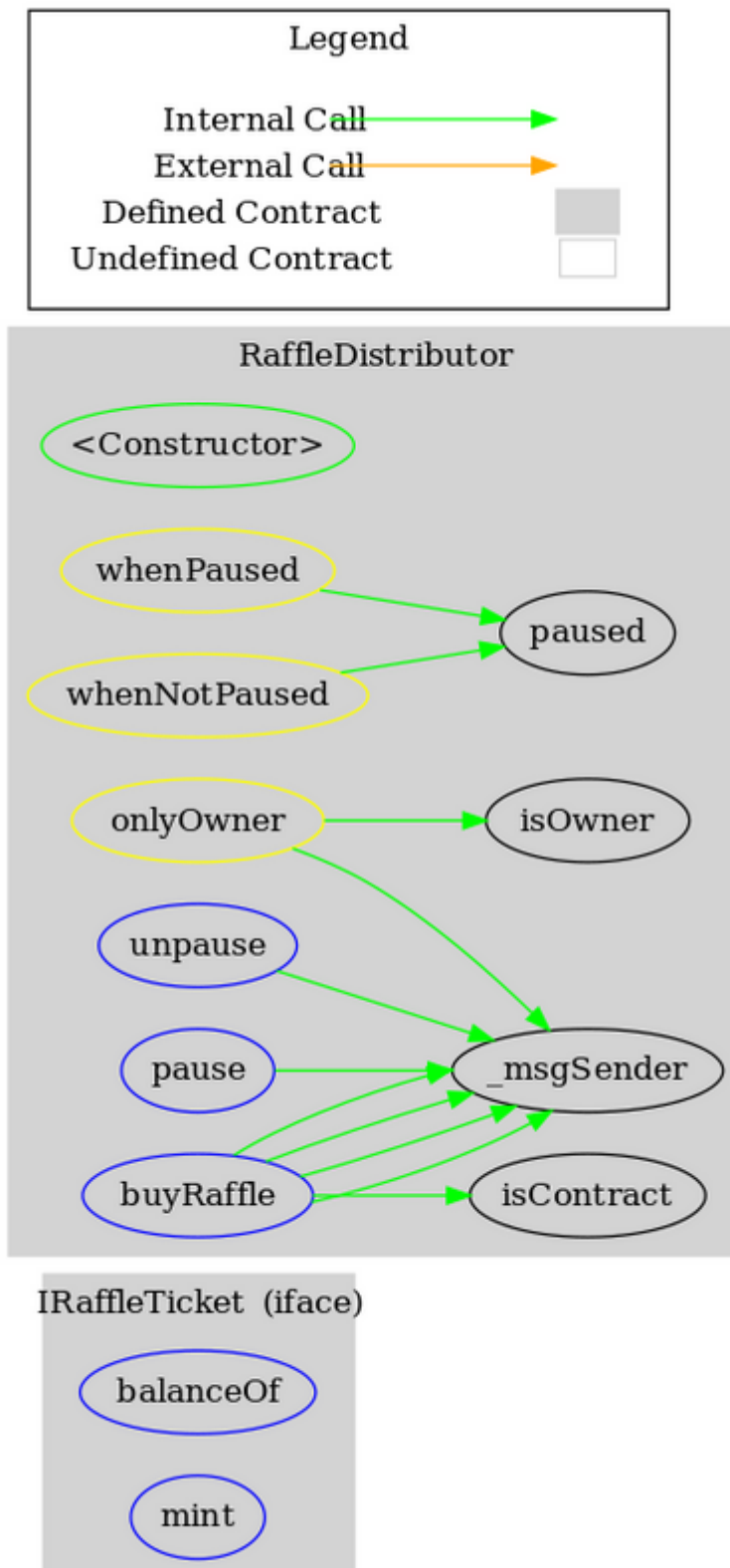
Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>.

# Contract Functions

| Contract                 | Type           | Bases      |            |               |
|--------------------------|----------------|------------|------------|---------------|
|                          | Function Name  | Visibility | Mutability | Modifiers     |
|                          |                |            |            |               |
| <b>IRaffleTicket</b>     | Interface      |            |            |               |
|                          | balanceOf      | External   |            | -             |
|                          | mint           | External   | ✓          | -             |
|                          |                |            |            |               |
| <b>RaffleDistributor</b> | Implementation |            |            |               |
|                          | <Constructor>  | Public     | ✓          | -             |
|                          | _msgSender     | Internal   |            |               |
|                          | isOwner        | Public     |            | -             |
|                          | paused         | Public     |            | -             |
|                          | pause          | External   | ✓          | onlyOwner     |
|                          | unpause        | External   | ✓          | onlyOwner     |
|                          | isContract     | Internal   |            |               |
|                          | buyRaffle      | External   | Payable    | whenNotPaused |

# Contract Flow



# Summary

The RaffleDistributor contract implements a ticket mechanism. It provides functionality to buy tickets. The audit investigates the main features, mentions security recommendation, performance improvements and potential optimizations.

# Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

# About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Cyberscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.



The Cyberscope team

<https://www.cyberscope.io>



Cyberscope

# Audit Report

## **RaffleRefund**

August 2022

SHA256    ddbf66cd6cfcad3aa98e3aaebc88139a63e8d2aa105ecf1cd8e8c93c7c36615d

Audited by   © cyberscope

# Table of Contents

|                                                         |           |
|---------------------------------------------------------|-----------|
| <b>Table of Contents</b>                                | <b>1</b>  |
| <b>Contract Review</b>                                  | <b>2</b>  |
| <b>Source Files</b>                                     | <b>2</b>  |
| <b>Audit Updates</b>                                    | <b>2</b>  |
| <b>Introduction</b>                                     | <b>3</b>  |
| <b>Request Verification</b>                             | <b>4</b>  |
| <b>Contract Roles</b>                                   | <b>5</b>  |
| <b>Contract Diagnostics</b>                             | <b>6</b>  |
| <b>USB - User Sufficient Balance</b>                    | <b>7</b>  |
| <b>Description</b>                                      | <b>7</b>  |
| <b>Recommendation</b>                                   | <b>7</b>  |
| <b>CSB - Contract Sufficient Balance</b>                | <b>8</b>  |
| <b>Description</b>                                      | <b>8</b>  |
| <b>Recommendation</b>                                   | <b>8</b>  |
| <b>L04 - Conformance to Solidity Naming Conventions</b> | <b>9</b>  |
| <b>Description</b>                                      | <b>9</b>  |
| <b>Recommendation</b>                                   | <b>9</b>  |
| <b>Contract Functions</b>                               | <b>10</b> |
| <b>Contract Flow</b>                                    | <b>11</b> |
| <b>Summary</b>                                          | <b>12</b> |
| <b>Disclaimer</b>                                       | <b>13</b> |
| <b>About Cyberscope</b>                                 | <b>14</b> |

## Contract Review

|                      |                                                                                                                                                                             |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Contract Name</b> | RaffleRefund                                                                                                                                                                |
| <b>Test Deploy</b>   | <a href="https://testnet.bscscan.com/address/0x83F4A44D061d8407fF2Dc408Ead0e563ca4C034d">https://testnet.bscscan.com/address/0x83F4A44D061d8407fF2Dc408Ead0e563ca4C034d</a> |
| <b>Domain</b>        | <a href="https://battleworld.game">https://battleworld.game</a>                                                                                                             |

## Source Files

|                     |                                                                  |
|---------------------|------------------------------------------------------------------|
| <b>Filename</b>     | <b>SHA256</b>                                                    |
| <b>contract.sol</b> | ddbf66cd6cfcad3aa98e3aaebc88139a63e8d2aa105ecf1cd8e8c93c7c36615d |

## Audit Updates

|                      |                 |
|----------------------|-----------------|
| <b>Initial Audit</b> | 4th August 2022 |
| <b>Corrected</b>     |                 |

# Introduction

The RaffleFund contract implements a ticket redeem mechanism. The users can redeem their tickets in order to receive native tokens. During the redeem process, the corresponding tickets are burned. The value of each ticket is defined by the contract owners during the contract deployment. During the redeem process, the users have to provide a message that must be verified in order to proceed with the transaction.

## Request Verification

The verification process is based on an off-chain configuration. The contract owners are responsible for updating the in-chain factor in order to validate correctly the provided message.

The verification algorithm is using the markle tree mechanism.

<https://github.com/protofire/zeppelin-solidity/blob/master/contracts/MerkleProof.sol>

According to the markle algorithm, the off-chain mechanism pre-defines all the `index, recipient, amount` combinations.

Hence, only predefined users have the ability to redeem tickets in specific amounts.

# Contract Roles

Role owner:

- The contract owners can pause the redeem mechanism.
- The contract owners can invalidate the validation factor and reset the saved address that claimed tickets.

# Contract Diagnostics

● Critical ● Medium ● Informative

| Severity | Code | Description                                |
|----------|------|--------------------------------------------|
| ●        | USB  | User Sufficient Balance                    |
| ●        | CSB  | Contract Sufficient Balance                |
| ●        | L04  | Conformance to Solidity Naming Conventions |

## USB - User Sufficient Balance

|                    |                   |
|--------------------|-------------------|
| <b>Criticality</b> | informative       |
| <b>Location</b>    | contract.sol#L297 |

### Description

The contract is baked on the fact that the burnFrom method will revert if the user's balance is insufficient.

```
require(  
    IRaffleTicket(RAFFLE_TICKET_ADDRESS).burnFrom(_msgSender(), amount),  
    "RaffleRefund#claimRaffleRefund: Burning Raffle Ticket Failed"  
);
```

### Recommendation

The contract could proactively check if the user's ticket balance is sufficient for the transaction.

## CSB - Contract Sufficient Balance

|                    |                   |
|--------------------|-------------------|
| <b>Criticality</b> | informative       |
| <b>Location</b>    | contract.sol#L301 |

### Description

The contract is based on the fact that the Vault contract will revert the transaction if the required balance is insufficient.

```
require(  
    IVault(VAULT).transferFromVault(  
        _msgSender(),  
        amount * RAFFLE_TICKET_PRICE  
    ),  
    "RaffleRefund#claimRaffleRefund: Payment from Vault Failed"  
);
```

### Recommendation

The contract could proactively check if the Vault's balance is sufficient

## L04 - Conformance to Solidity Naming Conventions

**Criticality**

minor

**Location**

contract.sol#L115,111,112,113

### Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow \_ at the beginning of the mixed\_case match for private variables and unused parameters.

```
RAFFLE_TICKET_PRICE  
VAULT  
RAFFLE_TICKET_ADDRESS  
RefundMerkleRoot
```

### Recommendation

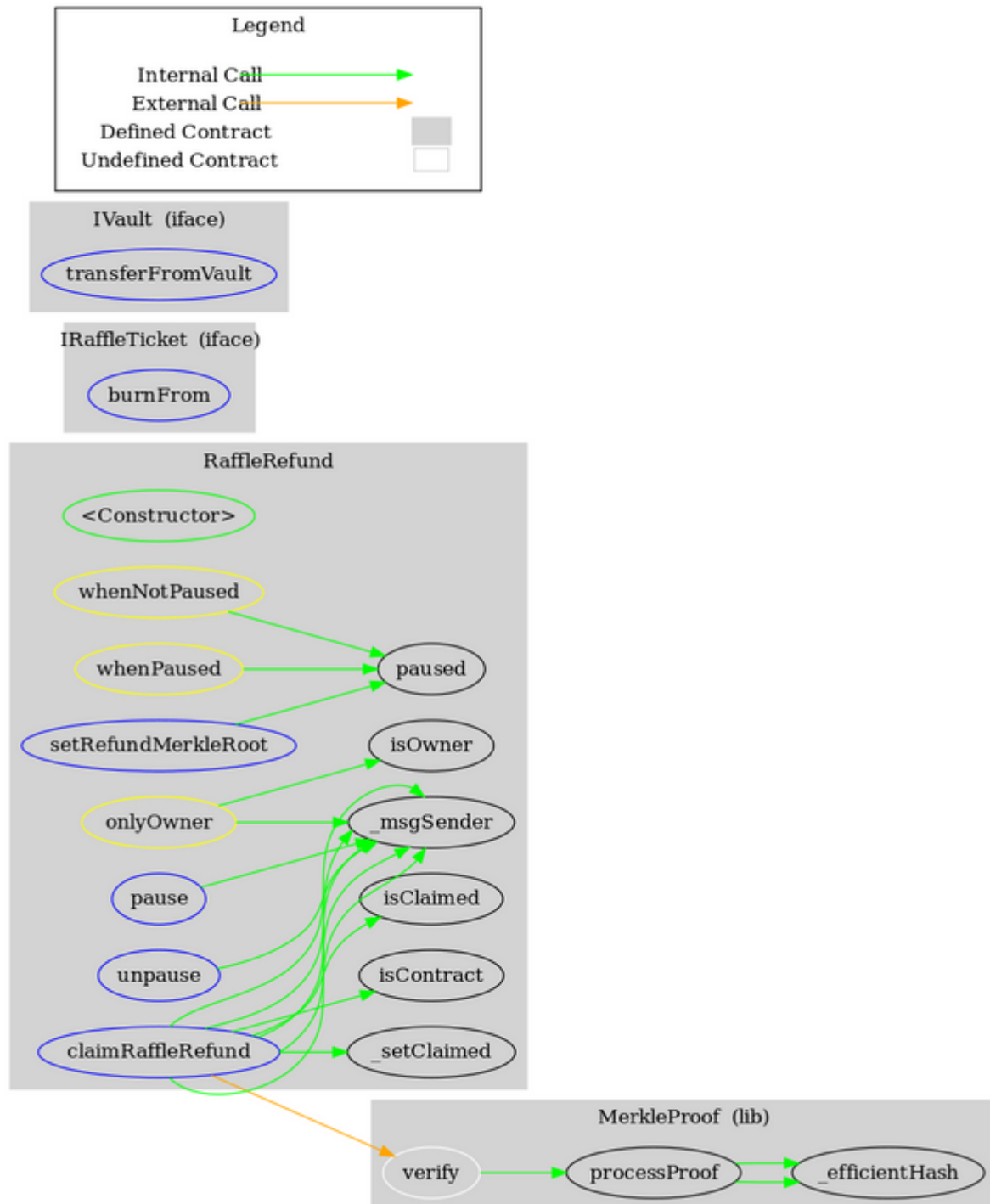
Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>.

# Contract Functions

| Contract             | Type                | Bases      |            |               |
|----------------------|---------------------|------------|------------|---------------|
|                      | Function Name       | Visibility | Mutability | Modifiers     |
|                      |                     |            |            |               |
| <b>MerkleProof</b>   | Library             |            |            |               |
|                      | verify              | Internal   |            |               |
|                      | processProof        | Internal   |            |               |
|                      | _efficientHash      | Private    |            |               |
|                      |                     |            |            |               |
| <b>IRaffleTicket</b> | Interface           |            |            |               |
|                      | burnFrom            | External   | ✓          | -             |
|                      |                     |            |            |               |
| <b>IVault</b>        | Interface           |            |            |               |
|                      | transferFromVault   | External   | ✓          | -             |
|                      |                     |            |            |               |
| <b>RaffleRefund</b>  | Implementation      |            |            |               |
|                      | <Constructor>       | Public     | ✓          | -             |
|                      | _msgSender          | Internal   |            |               |
|                      | isOwner             | Public     |            | -             |
|                      | paused              | Public     |            | -             |
|                      | pause               | External   | ✓          | onlyOwner     |
|                      | unpause             | External   | ✓          | onlyOwner     |
|                      | isContract          | Internal   |            |               |
|                      | setRefundMerkleRoot | External   | ✓          | onlyOwner     |
|                      | isClaimed           | Public     |            | -             |
|                      | _setClaimed         | Private    | ✓          |               |
|                      | claimRaffleRefund   | External   | ✓          | whenNotPaused |

# Contract Flow



## Summary

The Vault contract implements a redeem mechanism. It provides functionality to redeem tickets. The contract should thoroughly check balances before every transaction. The audit investigates the main features, mentions security recommendation, performance improvements and potential optimizations.

# Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

# About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.



The Cyberscope team

<https://www.cyberscope.io>



Cyberscope

# Audit Report

## **RaffleTicket**

August 2022

SHA256    6415fb263eac97432d8ca39fdf79b38c8b25a977ca27b8591a899a7b71e709dc

Audited by   © cyberscope

# Table of Contents

|                                                         |           |
|---------------------------------------------------------|-----------|
| <b>Table of Contents</b>                                | <b>1</b>  |
| <b>Contract Review</b>                                  | <b>2</b>  |
| <b>Source Files</b>                                     | <b>2</b>  |
| <b>Audit Updates</b>                                    | <b>2</b>  |
| <b>Introduction</b>                                     | <b>3</b>  |
| <b>Contract Roles</b>                                   | <b>4</b>  |
| <b>Contract Diagnostics</b>                             | <b>5</b>  |
| <b>L02 - State Variables could be Declared Constant</b> | <b>6</b>  |
| <b>Description</b>                                      | <b>6</b>  |
| <b>Recommendation</b>                                   | <b>6</b>  |
| <b>L04 - Conformance to Solidity Naming Conventions</b> | <b>7</b>  |
| <b>Description</b>                                      | <b>7</b>  |
| <b>Recommendation</b>                                   | <b>7</b>  |
| <b>Contract Functions</b>                               | <b>8</b>  |
| <b>Contract Flow</b>                                    | <b>9</b>  |
| <b>Summary</b>                                          | <b>10</b> |
| <b>Disclaimer</b>                                       | <b>11</b> |
| <b>About Cyberscope</b>                                 | <b>12</b> |

## Contract Review

|                       |                                                                                                                                                                             |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Contract Name</b>  | RaffleTicket                                                                                                                                                                |
| <b>Testing Deploy</b> | <a href="https://testnet.bscscan.com/address/0x869F7B1c9961d8eA04fCFDdc0AD556A8Af33e121">https://testnet.bscscan.com/address/0x869F7B1c9961d8eA04fCFDdc0AD556A8Af33e121</a> |
| <b>Symbol</b>         | RFT                                                                                                                                                                         |
| <b>Domain</b>         | <a href="https://battleworld.game">https://battleworld.game</a>                                                                                                             |

## Source Files

|                     |                                                                  |
|---------------------|------------------------------------------------------------------|
| <b>Filename</b>     | <b>SHA256</b>                                                    |
| <b>contract.sol</b> | 6415fb263eac97432d8ca39fdf79b38c8b25a977ca27b8591a899a7b71e709dc |

## Audit Updates

|                      |                 |
|----------------------|-----------------|
| <b>Initial Audit</b> | 4th August 2022 |
| <b>Corrected</b>     |                 |

# Introduction

The RaffleTicket contract implements a digital currency for the Battle World ecosystem. The contract owner can authorize or unauthorize addresses for mint and burn functionality. The contract owners have the ability to mint and burn tickets. The mintAccess authorized users can mint tickets. The burnAccess authorized users can burn tickets.

# Contract Roles

Role owner:

- The contract owners can authorize burn tickets.
- The contract owners can authorize mint tickets.
- The contract owners can mint and burn tickets.

Role mintAccess :

- The authorized addresses have the ability to mint tickets.

Role burnAccess :

- The authorized addresses have the ability to burn tickets.

# Contract Diagnostics

● Critical    ● Medium    ● Minor

| Severity | Code | Description                                |
|----------|------|--------------------------------------------|
| ●        | L02  | State Variables could be Declared Constant |
| ●        | L04  | Conformance to Solidity Naming Conventions |

## L02 - State Variables could be Declared Constant

**Criticality**

minor

**Location**

contract.sol#L26

### Description

Constant state variables should be declared constant to save gas.

decimals

### Recommendation

Add the constant attribute to state variables that never change.

## L04 - Conformance to Solidity Naming Conventions

**Criticality**

minor

**Location**

contract.sol#L30,31

### Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow \_ at the beginning of the mixed\_case match for private variables and unused parameters.

```
BurnAccess  
MintAccess
```

### Recommendation

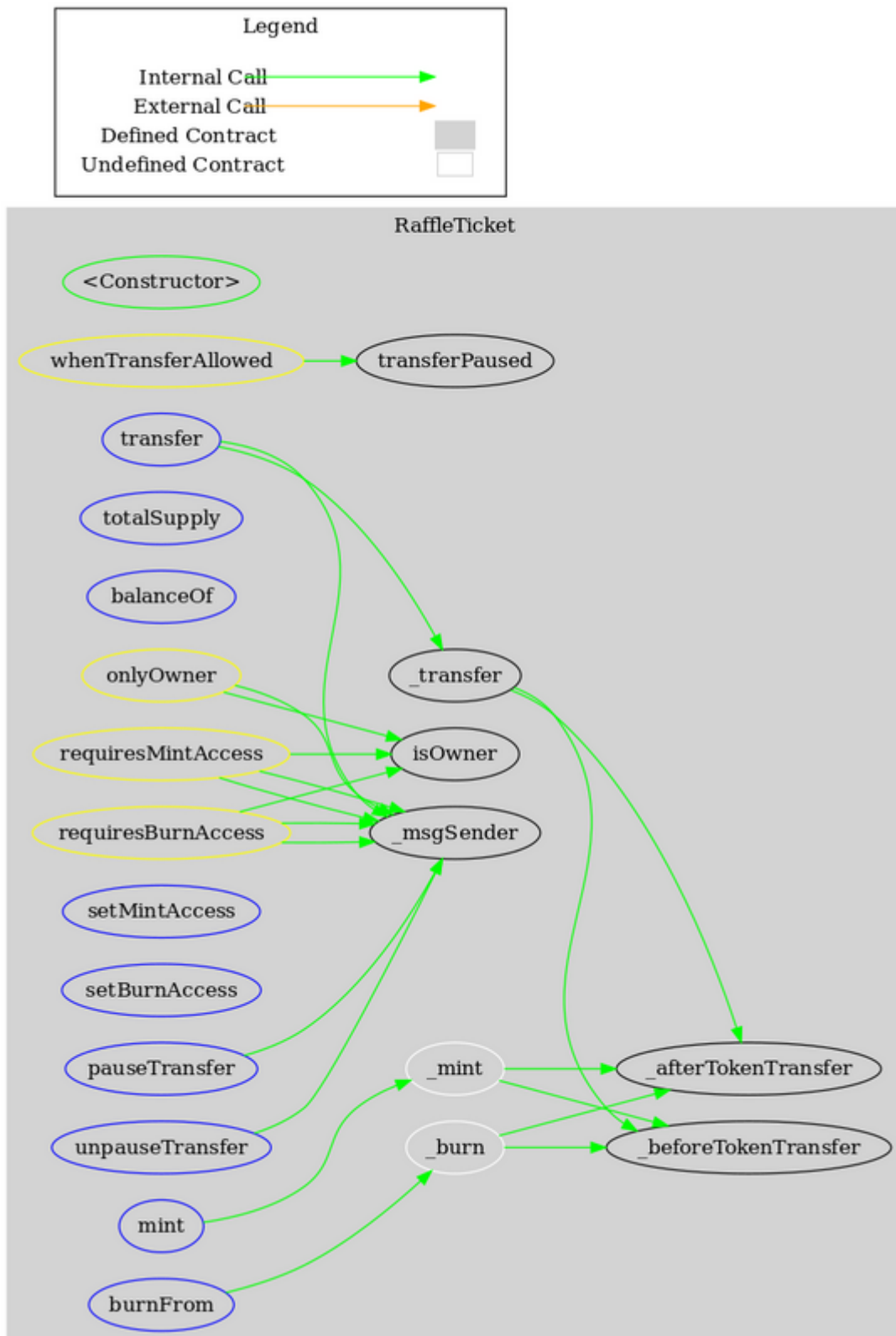
Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>.

# Contract Functions

| Contract            | Type                 | Bases      |            |                     |
|---------------------|----------------------|------------|------------|---------------------|
|                     | Function Name        | Visibility | Mutability | Modifiers           |
|                     |                      |            |            |                     |
| <b>RaffleTicket</b> | Implementation       |            |            |                     |
|                     | <Constructor>        | Public     | ✓          | -                   |
|                     | _msgSender           | Internal   |            |                     |
|                     | isOwner              | Public     |            | -                   |
|                     | transferPaused       | Public     |            | -                   |
|                     | totalSupply          | External   |            | -                   |
|                     | balanceOf            | External   |            | -                   |
|                     | transfer             | External   | ✓          | -                   |
|                     | _transfer            | Internal   | ✓          | whenTransferAllowed |
|                     | _mint                | Internal   | ✓          |                     |
|                     | _burn                | Internal   | ✓          |                     |
|                     | _beforeTokenTransfer | Internal   | ✓          |                     |
|                     | _afterTokenTransfer  | Internal   | ✓          |                     |
|                     | setMintAccess        | External   | ✓          | onlyOwner           |
|                     | setBurnAccess        | External   | ✓          | onlyOwner           |
|                     | pauseTransfer        | External   | ✓          | onlyOwner           |
|                     | unpauseTransfer      | External   | ✓          | onlyOwner           |
|                     | mint                 | External   | ✓          | requiresMintAccess  |
|                     | burnFrom             | External   | ✓          | requiresBurnAccess  |

# Contract Flow



## Summary

The RaffleTicket contract implements a digital currency. It provides functionality to burn and mint tickets to address. The audit investigates the main features, mentions security recommendation, performance improvements and potential optimizations.

## Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

# About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.



The Cyberscope team

<https://www.cyberscope.io>



Cyberscope

# Audit Report

## **Vault**

August 2022

SHA256 7470f1b9e918c1d4cbc18846b94537bbf1629b6310c242b7db99e5be1b86fad7

Audited by © cyberscope

# Table of Contents

|                                                         |           |
|---------------------------------------------------------|-----------|
| <b>Table of Contents</b>                                | <b>1</b>  |
| <b>Contract Review</b>                                  | <b>2</b>  |
| <b>Source Files</b>                                     | <b>2</b>  |
| <b>Audit Updates</b>                                    | <b>2</b>  |
| <b>Introduction</b>                                     | <b>3</b>  |
| <b>Contract Roles</b>                                   | <b>4</b>  |
| <b>Contract Diagnostics</b>                             | <b>5</b>  |
| <b>L04 - Conformance to Solidity Naming Conventions</b> | <b>6</b>  |
| <b>Description</b>                                      | <b>6</b>  |
| <b>Recommendation</b>                                   | <b>6</b>  |
| <b>Contract Functions</b>                               | <b>7</b>  |
| <b>Contract Flow</b>                                    | <b>8</b>  |
| <b>Summary</b>                                          | <b>9</b>  |
| <b>Disclaimer</b>                                       | <b>10</b> |
| <b>About Cyberscope</b>                                 | <b>11</b> |

## Contract Review

|                       |                                                                                                                                                                             |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Contract Name</b>  | Vault                                                                                                                                                                       |
| <b>Testing Deploy</b> | <a href="https://testnet.bscscan.com/address/0xd1c8d2cf5AE112d633174bc73F4Ad7E55414B769">https://testnet.bscscan.com/address/0xd1c8d2cf5AE112d633174bc73F4Ad7E55414B769</a> |
| <b>Domain</b>         | <a href="https://battleworld.game">https://battleworld.game</a>                                                                                                             |

## Source Files

|                     |                                                                  |
|---------------------|------------------------------------------------------------------|
| <b>Filename</b>     | <b>SHA256</b>                                                    |
| <b>contract.sol</b> | 7470f1b9e918c1d4cbc18846b94537bbf1629b6310c242b7db99e5be1b86fad7 |

## Audit Updates

|                      |                 |
|----------------------|-----------------|
| <b>Initial Audit</b> | 4th August 2022 |
| <b>Corrected</b>     |                 |

# Introduction

The Vault contract implements a vault safer functionality for native tokens. The contract owner controls the access to the contract functionality, by authorize or unauthorize addresses to transfer balance from the Vault. The contract owners have the ability to stop the withdrawal operation for all the authorized addresses. The contract owners have the ability to withdraw all the funds of the vault. The authorized addresses have the ability to transfer all the funds to any address. There is no transfer limit for both owners and authorized addresses.

# Contract Roles

Role owner:

- The contract owners can authorize withdrawal addresses.
- The contract owners can pause the withdrawals.
- The contract owners can withdraw.

Role authorized :

- The authorized users can withdraw.

# Contract Diagnostics

● Critical    ● Medium    ● Minor

| Severity | Code | Description                                |
|----------|------|--------------------------------------------|
| ●        | L04  | Conformance to Solidity Naming Conventions |

## L04 - Conformance to Solidity Naming Conventions

|                    |                   |
|--------------------|-------------------|
| <b>Criticality</b> | minor             |
| <b>Location</b>    | contract.sol#L150 |

### Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow \_ at the beginning of the mixed\_case match for private variables and unused parameters.

OwnerWithdraw

### Recommendation

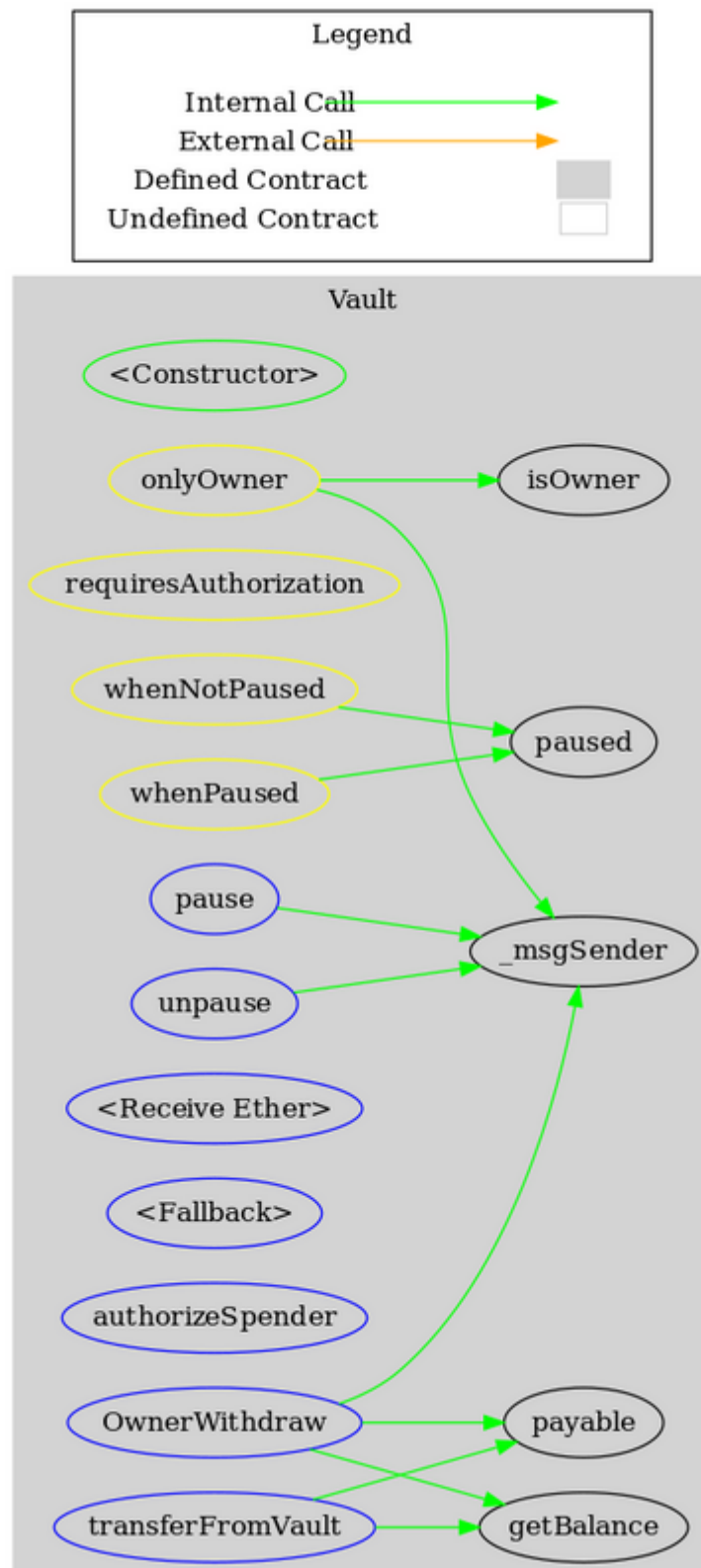
Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>.

# Contract Functions

| Contract | Type              | Bases      |            |                                        |
|----------|-------------------|------------|------------|----------------------------------------|
|          | Function Name     | Visibility | Mutability | Modifiers                              |
|          |                   |            |            |                                        |
| Vault    | Implementation    |            |            |                                        |
|          | <Constructor>     | Public     | ✓          | -                                      |
|          | _msgSender        | Internal   |            |                                        |
|          | isOwner           | Public     |            | -                                      |
|          | paused            | Public     |            | -                                      |
|          | pause             | External   | ✓          | onlyOwner                              |
|          | unpause           | External   | ✓          | onlyOwner                              |
|          | <Receive Ether>   | External   | Payable    | -                                      |
|          | <Fallback>        | External   | Payable    | -                                      |
|          | getBalance        | Public     |            | -                                      |
|          | authorizeSpender  | External   | ✓          | onlyOwner                              |
|          | OwnerWithdraw     | External   | ✓          | onlyOwner                              |
|          | transferFromVault | External   | ✓          | whenNotPaused<br>requiresAuthorization |

# Contract Flow



# Summary

The Vault contract implements a vault. It provides functionality to withdraw and transfer funds. The audit investigates the main features, mentions security recommendation, performance improvements and potential optimizations.

## Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

# About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.



The Cyberscope team

<https://www.cyberscope.io>