# Smart Contract Security Audit V1

## Intercoin Investor

https://intercoin.org/

19/11/2021

https://saferico.com/

business@saferico.com
https://t.me/SFI_ANN
–

# 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

- **Website**: https://intercoin.org/
- **Twitter**: https://twitter.com/IntercoinOrg
- **Telegram group:** https://t.me/intercoin
- **Whitepaper:** https://intercoin.org/whitepaper.pdf
- **Coingecko:** https://www.coingecko.com/en/coins/intercoin
- **Reddit:** https://www.reddit.com/r/intercoin/
- **Facebook**: https://www.facebook.com/intercoin/
- **GitHub:** https://github.com/Intercoin
- **LinkedIn:** https://www.linkedin.com/company/intercoin-inc/
- **Platform**: Ethereum Network
- **Contract Address**: 0x1111158f88410da5f92c7e34c01e7b8649bc0155

**Intercoin (ITR)** is a blockchain-based purpose-driven company with cryptographic tools to introduce new systems to significantly improve people's experience with connections and the next steps in the evolution of money.

## Token Information

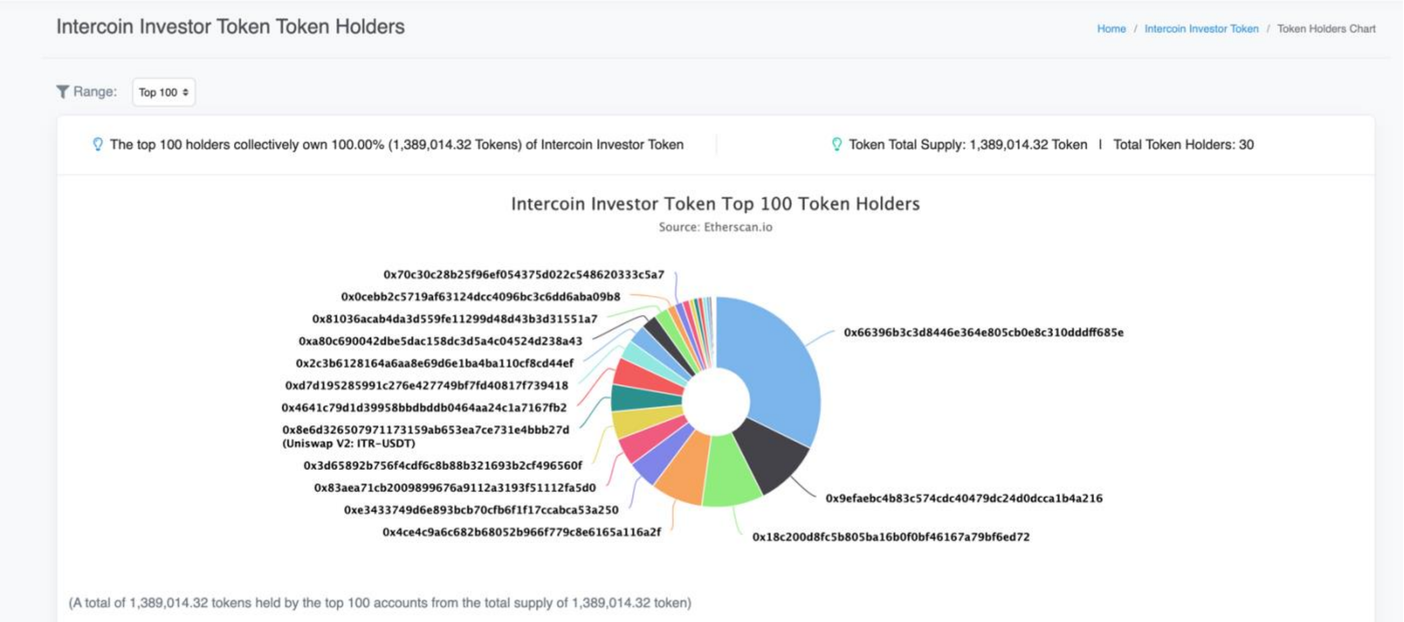
- Name: ITR
- Total Supply: 1,389,014.32
- Holders: 30 address
- Total transactions: 92

## Contract address which this token depends on it:

Intercoin (ITR) contract on ETH main net

https://etherscan.io/address/0x6Ef5febbD2A56FAb23f18a69d3fB9F4E2A70440B

# ITR Token Distribution



Intercoin Investor Token Token Holders

Home / Intercoin Investor Token / Token Holders Chart

Range: Top 100

The top 100 holders collectively own 100.00% (1,389,014.32 Tokens) of Intercoin Investor Token

Token Total Supply: 1,389,014.32 Token | Total Token Holders: 30

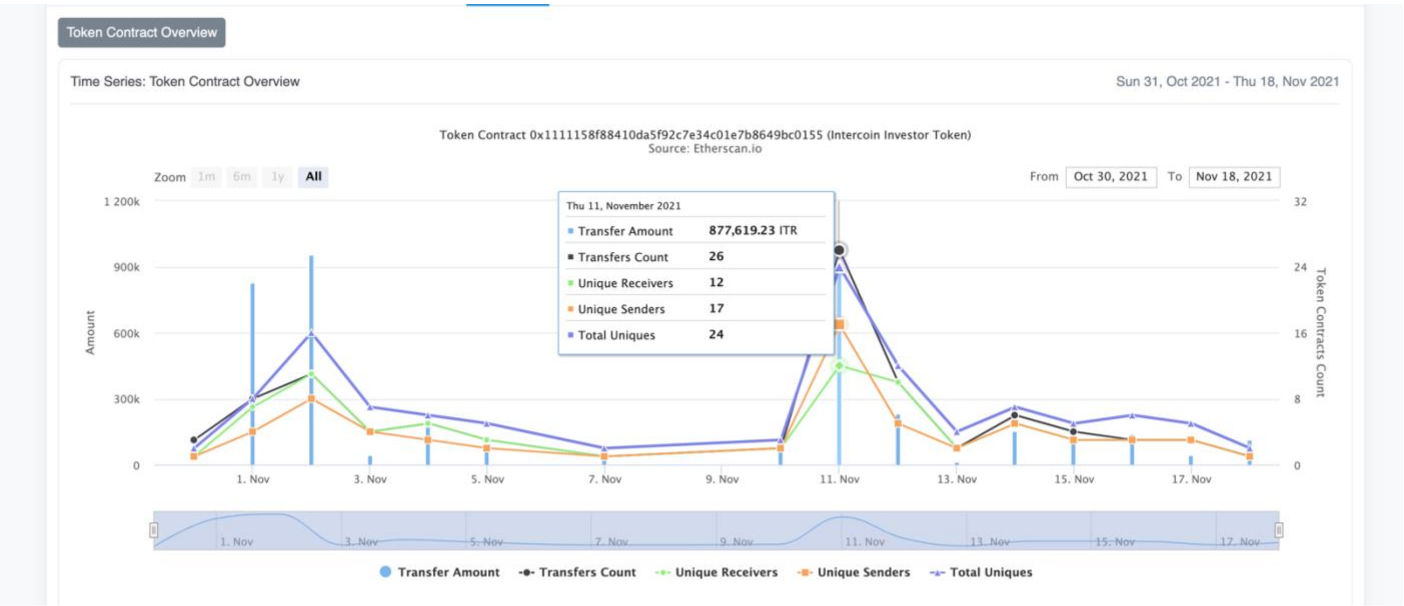
Intercoin Investor Token Top 100 Token Holders

Source: Etherscan.io

0x70c30c28b25f96ef054375d022c548620333c5a7
0x0cebb2c5719af63124dcc4096bc3c6dd6aba09b8
0x81036acab4da3d559fe11299d48d43b3d31551a7
0xa80c690042dbe5dac158dc3d5a4c04524d238a43
0x2c3b6128164a6aa8e69d6e1ba4ba110cf8cd44ef
0xd7d195285991c276e427749bf7fd40817f739418
0x4641c79d1d39958bbdbddb0464aa24c1a7167fb2
0x8e6d326507971173159ab653ea7ce731e4bbb27d (Uniswap V2: ITR-USDT)
0x3d65892b756f4cdf6c8b88b321693b2cf496560f
0x83aea71cb2009899676a9112a3193f51112fa5d0
0xe3433749d6e893bcb70cfb6f1f17ccabca53a250
0x4ce4c9a6c682b68052b966f779c8e6165a116a2f
0x66396b3c3d8446e364e805cb0e8c310dddff685e
0x9efaebc4b83c574cdc40479dc24d0dcca1b4a216
0x18c200d8fc5b805ba16b0f0bf46167a79bf6ed72

(A total of 1,389,014.32 tokens held by the top 100 accounts from the total supply of 1,389,014.32 token)

# Contract Interaction Details



Token Contract Overview

Time Series: Token Contract Overview

Sun 31, Oct 2021 - Thu 18, Nov 2021

Token Contract 0x1111158f88410da5f92c7e34c01e7b8649bc0155 (Intercoin Investor Token)

Source: Etherscan.io

Thu 11, November 2021

- Transfer Amount 877,619.23 ITR
- Transfers Count 26
- Unique Receivers 12
- Unique Senders 17
- Total Uniques 24

# Executive Summary

According to our assessment, the customer`s solidity smart contract is **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, 2 very low-level issues and 3 notes in all solidity files of the contract

The files:

ITR.sol

# File and Function Level Report

## File in Scope:

| Contract Name | SHA 256 hash | Contract Address |
|---|---|---|
| ITR.sol | **56823854ccefa1ed38259901f4a8e500add3a9d8a65459e369b01f25e1ee110b** | 0x1111158f88410da5f92c7e34c01e7b8649bc0155 |

- Contract: ITR
- Inherit: Ownable, ERC777
- Observation: All passed including security check
- Test Report: passed
- Score: passed
- Conclusion: passed

| Function | Test Result | Type / Return Type | Score |
|---|---|---|---|
| name | ✓ | Read / public | **Passed** |
| symbol | ✓ | Read / public | **Passed** |
| decimals | ✓ | Read / public | **Passed** |
| totalSupply | ✓ | Read / public | **Passed** |
| allowance | ✓ | Read / public | **Passed** |
| balanceOf | ✓ | Read / public | **Passed** |
| Owner | ✓ | Read / public | **Passed** |
| totalRemaining | ✓ | Read / public | **Passed** |
| isOperatorFor | ✓ | Read / public | **Passed** |
| granularity | ✓ | Read / public | **Passed** |

| | | | |
|---|---|---|---|
| defaultOperators | ✓ | Read / public | Passed |
| getMaxTotalSupply | ✓ | Read / public | Passed |
| getClaimFraction | ✓ | Read / public | Passed |
| _claimDuration | ✓ | Read / private | Passed |
| _claimFraction | ✓ | Read / private | Passed |
| approve | ✓ | Write / public | Passed |
| TransferFrom | ✓ | Write / public | Passed |
| send | ✓ | Write / public | Passed |
| transfer | ✓ | Write / public | Passed |
| burn | ✓ | Write / public | Passed |
| claim | ✓ | Write / public | Passed |
| authorizeOperator | ✓ | Write / public | Passed |
| revokeOperator | ✓ | Write / public | Passed |
| operatorSend | ✓ | Write / public | Passed |
| operatorBurn | ✓ | Write / public | Passed |
| renounceOwnership | ✓ | Write / public | Passed |
| transferOwnership | ✓ | Write / public | Passed |

# Issues Checking Status

| No. | Issue Description | Checking Status |
|---|---|---|
| 1 | Compiler warnings. | Passed |
| 2 | Race conditions and Reentrancy. Cross-function race conditions. | Passed |
| 3 | Possible delays in data delivery. | Passed |
| 4 | Oracle calls. | Passed |
| 5 | Front running. | Passed |
| 6 | Timestamp dependence. | Passed |
| 7 | Integer Overflow and Underflow. | Passed |
| 8 | DoS with Revert. | Passed |
| 9 | DoS with block gas limit. | Passed |
| 10 | Methods execution permissions. | Passed |
| 11 | Economy model. If application logic is based on an incorrect economic model, the application would not function correctly and participants would incur financial losses.<br>This type of issue is most often found in bonus rewards systems, Staking and Farming contracts, Vault and Vesting contracts, etc. | Passed |
| 12 | The impact of the exchange rate on the logic. | Passed |
| 13 | Private user data leaks. | Passed |
| 14 | Malicious Event log. | Passed |
| 15 | Scoping and Declarations. | Passed |
| 16 | Uninitialized storage pointers. | Passed |
| 17 | Arithmetic accuracy. | Passed |
| 18 | Design Logic. | 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:

### Issue #1.

In detail Inline assembly:

The Contract uses inline assembly, this is only advised in rare cases.
Additionally, static analysis modules do not parse inline Assembly, this can lead to wrong analysis results.

```
assembly {
            size := extcodesize(account)
        }
      assembly {
                    let returndata_size := mload(returndata)
                    revert(add(32, returndata), returndata_size)
                }
```

### Issue #2.

In detail Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree.
That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

```
deployedTime = block.timestamp;
uint256 index = (block.timestamp)
return (block.timestamp.sub(deployedTime)).div(_claimDuration);
```

## Very Low:

### Issue #1. For loop over a dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point.

Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

```
for (uint256 i = 0; i < defaultOperators_.length; i++) {
            _defaultOperators[defaultOperators_[i]] = true;
        }
```

### Issue #2. Low level calls:

In detail Use of "call": should be avoided whenever possible.It can lead to unexpected behavior if return value is not handled properly.
Please use Direct Calls via specifying the called contract's interface.

```
(bool success, ) = recipient.call{value: amount}("");
(bool success, bytes memory returndata) = target.call{value:
value}(data);
(bool success, bytes memory returndata) = target.delegatecall(data);
```

## Notes:

### #Note1 Similar variable names:

ERC777.(string,string,address[]) : Variables have very similar names "_name", symbol and "name_", symbol.

```
_name = name_;
_symbol = symbol_;
```

### #Note2 Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

```
require(owner() == _msgSender(), "Ownable: caller is not the owner");
require(newOwner != address(0), "Ownable: new owner is the zero
address");
```

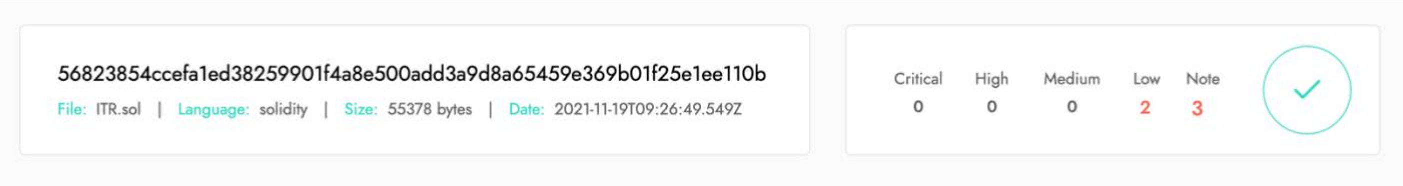
### #Note3 Delete from the dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.

```
delete _revokedDefaultOperators[_msgSender()][operator];
delete _operators[_msgSender()][operator];
```

# Automatic Testing

## 1-Check for security

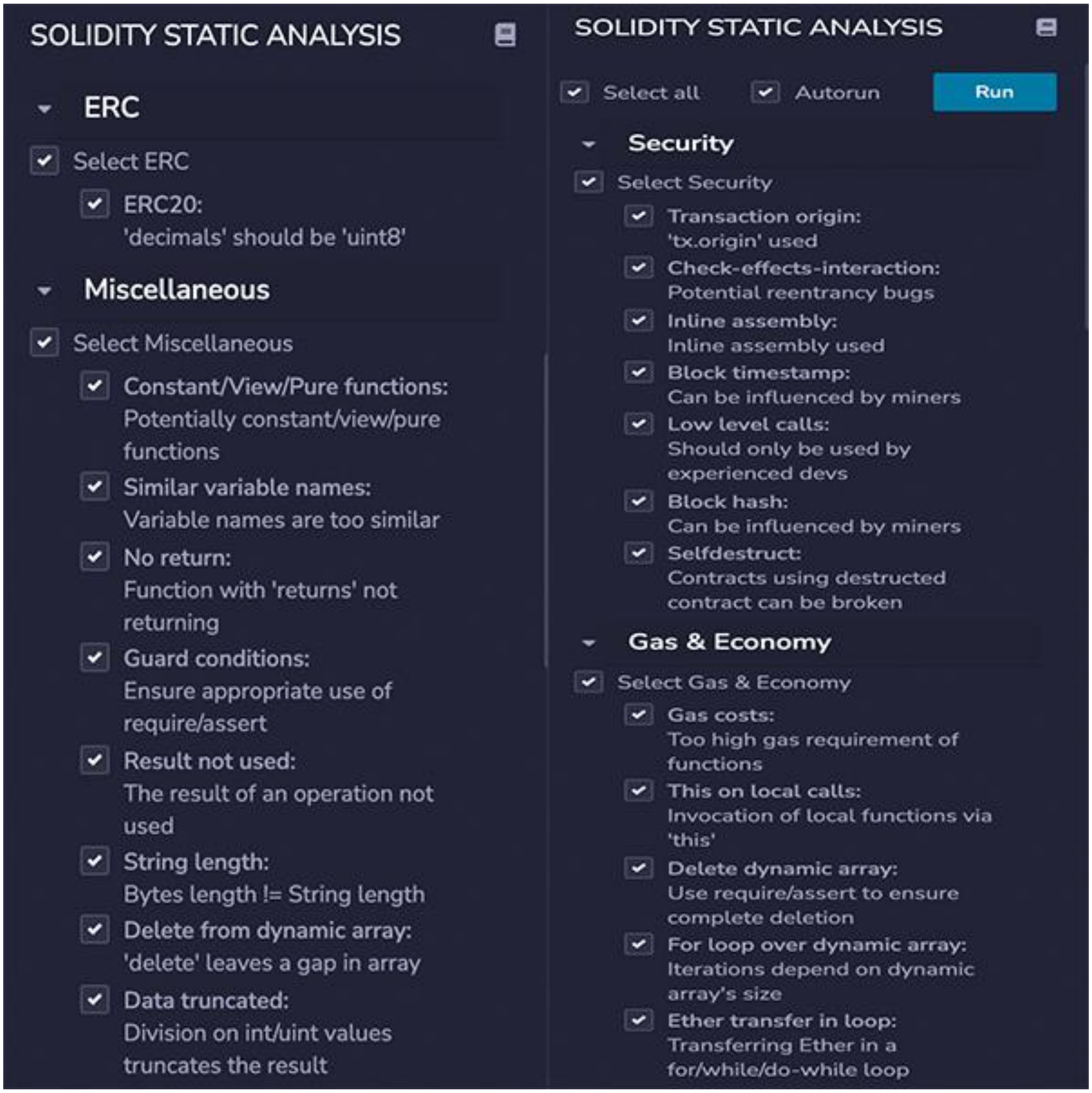


56823854ccefa1ed38259901f4a8e500add3a9d8a65459e369b01f25e1ee110b

File: ITR.sol | Language: solidity | Size: 55378 bytes | Date: 2021-11-19T09:26:49.549Z

| Critical | High | Medium | Low | Note |
|---|---|---|---|---|
| 0 | 0 | 0 | 2 | 3 |

## 2- SOLIDITY STATIC ANALYSIS



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

SOLIDITY STATIC ANALYSIS

☑ Select all ☑ Autorun Run

Security

☑ Select Security

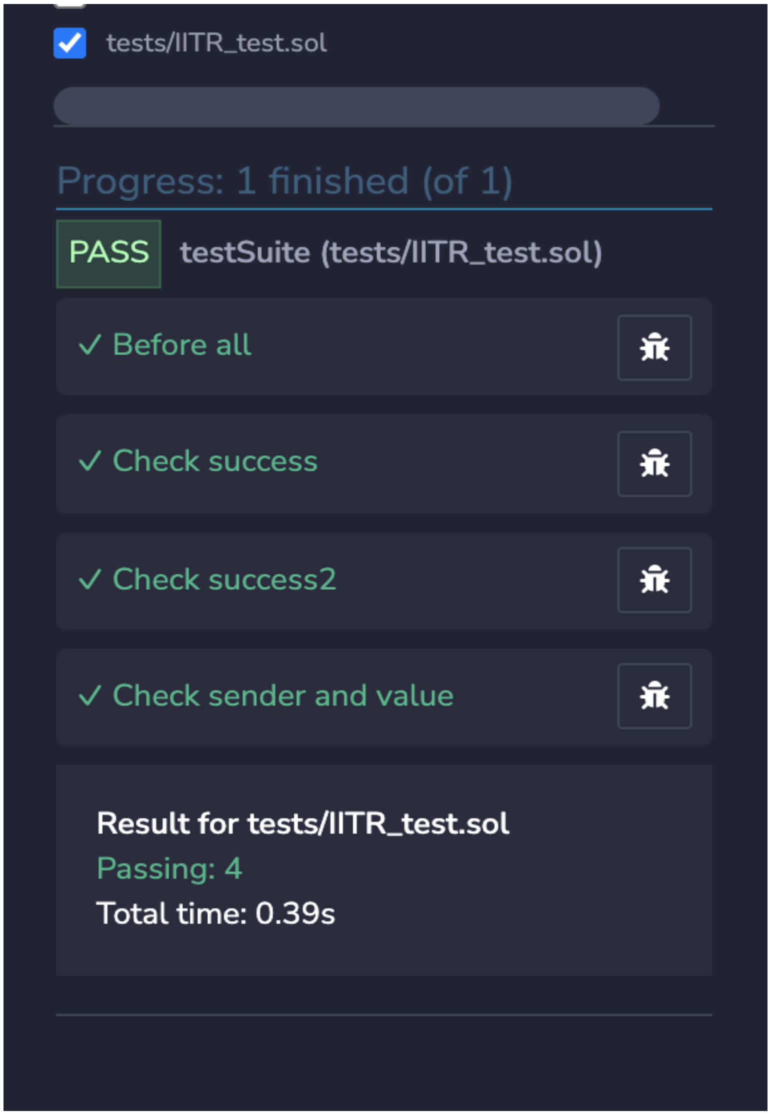
- ☑ Transaction origin: 'tx.origin' used
- ☑ Check-effects-interaction: Potential reentrancy bugs
- ☑ Inline assembly: Inline assembly used
- ☑ Block timestamp: Can be influenced by miners
- ☑ Low level calls: Should only be used by experienced devs
- ☑ Block hash: Can be influenced by miners
- ☑ Selfdestruct: 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

3- SOLIDITY UNIT TESTING



☑ tests/IITR_test.sol

Progress: 1 finished (of 1)

PASS testSuite (tests/IITR_test.sol)

✓ Before all

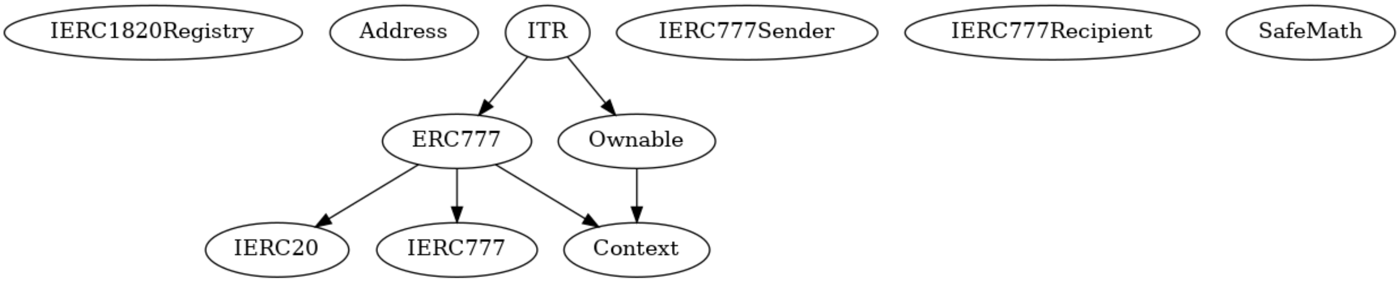
✓ Check success

✓ Check success2

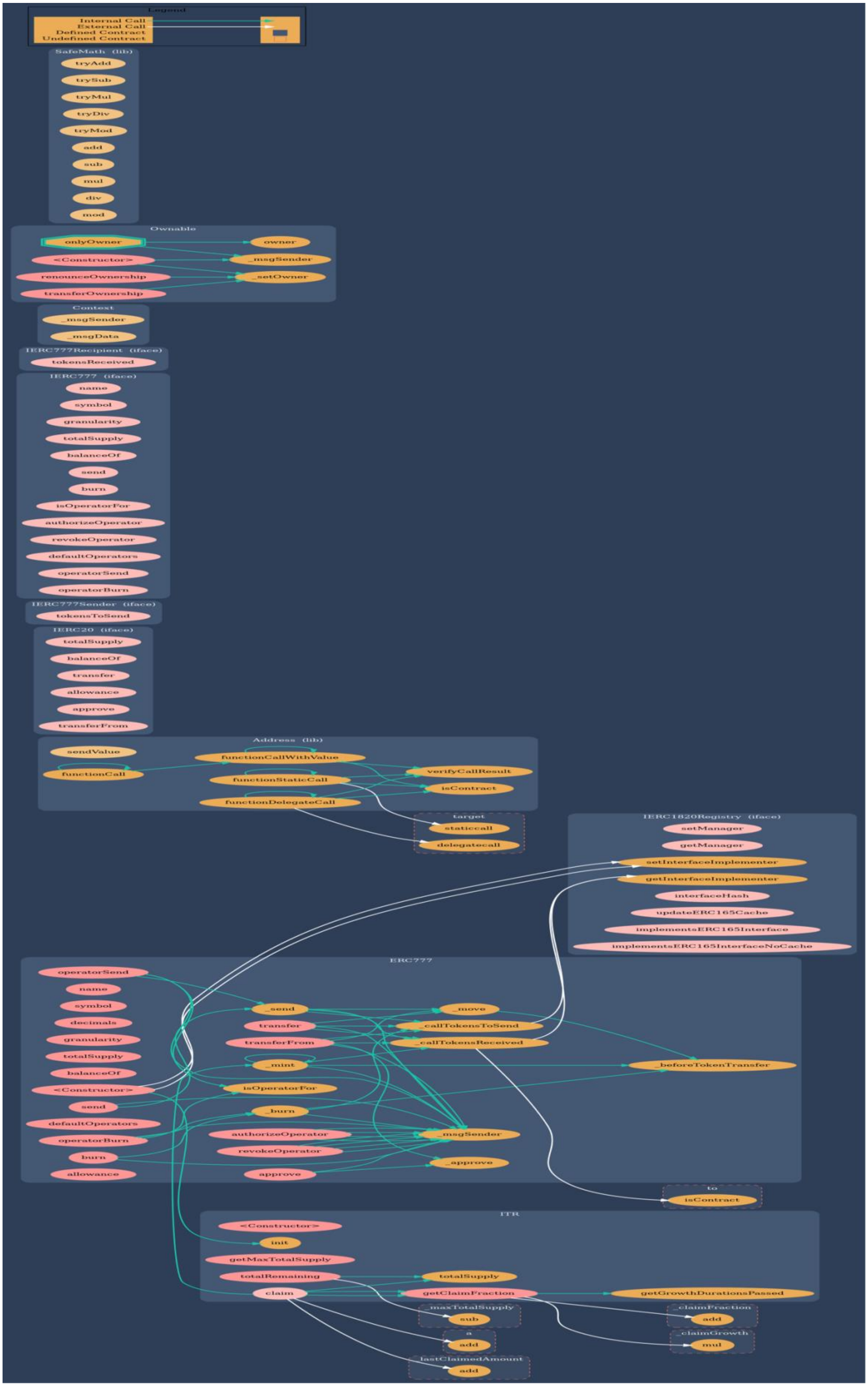
✓ Check sender and value

Result for tests/IITR_test.sol
Passing: 4
Total time: 0.39s

# #Inheritance graph



# #Call graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
SafeMath (lib)
tryAdd
trySub
tryMul
tryDiv
tryMod
add
sub
mul
div
mod
Ownable
onlyOwner
owner
<Constructor>
_msgSender
renounceOwnership
_setOwner
transferOwnership
Context
_msgSender
_msgData
IERC777Recipient (iface)
tokensReceived
IERC777 (iface)
name
symbol
granularity
totalSupply
balanceOf
send
burn
isOperatorFor
authorizeOperator
revokeOperator
defaultOperators
operatorSend
operatorBurn
IERC777Sender (iface)
tokensToSend
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
Address (lib)
sendValue
functionCallWithValue
functionCall
verifyCallResult
functionStaticCall
isContract
functionDelegateCall
target
staticcall
delegatecall
IERC1820Registry (iface)
setManager
getManager
setInterfaceImplementer
getInterfaceImplementer
interfaceHash
updateERC165Cache
implementsERC165Interface
implementsERC165InterfaceNoCache
ERC777
operatorSend
name
symbol
decimals
granularity
totalSupply
balanceOf
<Constructor>
send
defaultOperators
operatorBurn
burn
allowance
_send
transfer
transferFrom
_mint
isOperatorFor
_burn
authorizeOperator
revokeOperator
approve
_move
_callTokensToSend
_callTokensReceived
_beforeTokenTransfer
_msgSender
_approve
to
isContract
ITR
<Constructor>
init
getMaxTotalSupply
totalRemaining
totalSupply
claim
getClaimFraction
getGrowthDurationsPassed
_maxTotalSupply
sub
_claimFraction
add
a
add
_claimGrowth
mul
lastClaimedAmount
add

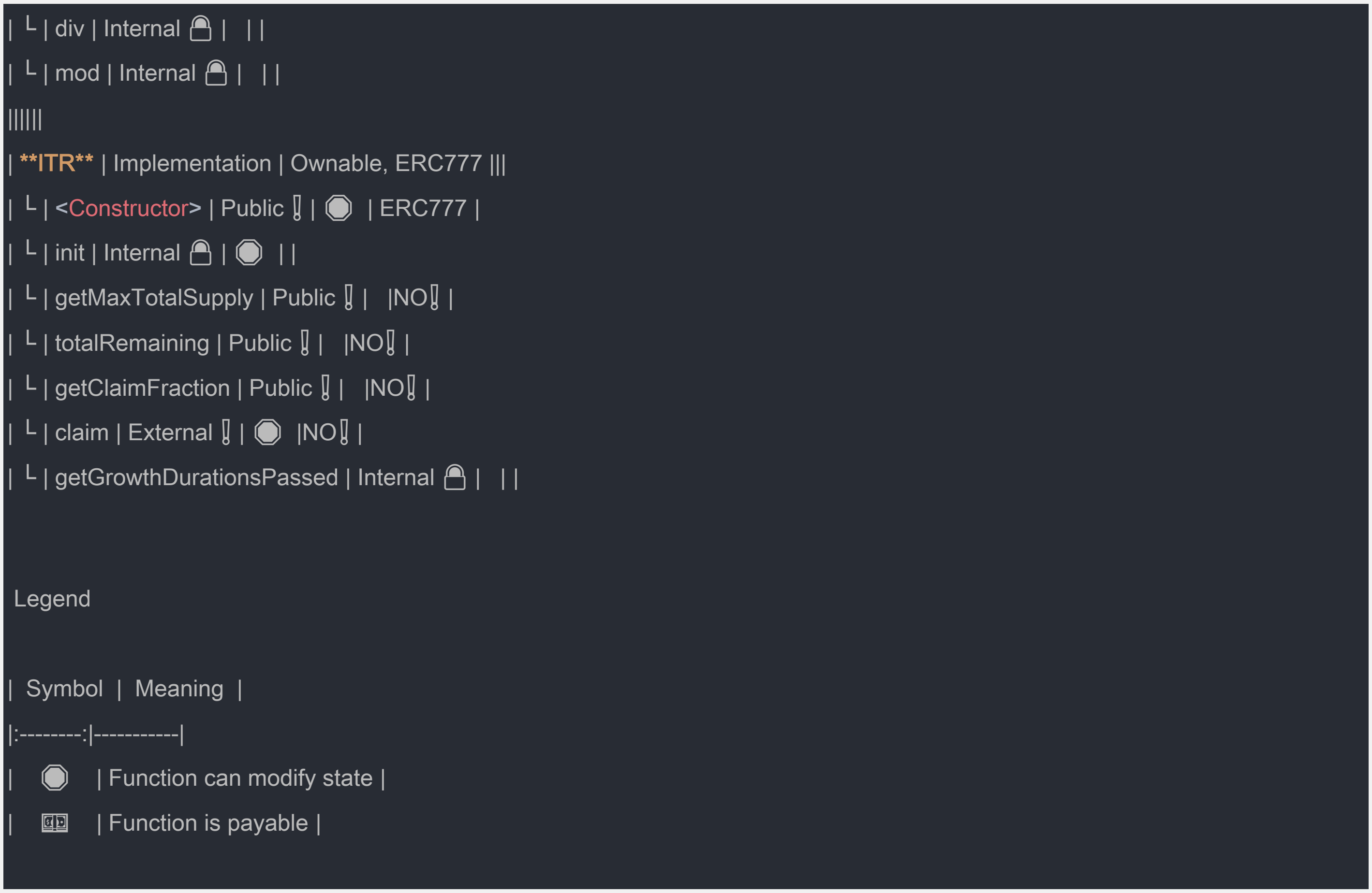
# Automatic general report

```
Files Description Table


|  File Name  |  SHA-1 Hash  |
|-------------|--------------|
| /Users/macbook/Desktop/smart contracts/ITR.sol | 72fd2633f96df518117b774c390bc8669b484e70 |


 Contracts Description Table


|  Contract  |        Type        |       Bases      |                  |                 |
|:----------:|:-------------------:|:----------------:|:----------------:|:---------------:|
|     └      |  **Function Name**  |  **Visibility**  |  **Mutability**  |  **Modifiers**  |
||||||
| **IERC1820Registry** | Interface |  |||
| └ | setManager | External ❗ | 🛑  |NO❗ |
| └ | getManager | External ❗ |   |NO❗ |
| └ | setInterfaceImplementer | External ❗ | 🛑  |NO❗ |
| └ | getInterfaceImplementer | External ❗ |   |NO❗ |
| └ | interfaceHash | External ❗ |   |NO❗ |
| └ | updateERC165Cache | External ❗ | 🛑  |NO❗ |
| └ | implementsERC165Interface | External ❗ |   |NO❗ |
| └ | implementsERC165InterfaceNoCache | External ❗ |   |NO❗ |
||||||
| **Address** | Library |  |||
| └ | isContract | Internal 🔒 |   | |
| └ | sendValue | Internal 🔒 | 🛑  | |
| └ | functionCall | Internal 🔒 | 🛑  | |
| └ | functionCall | Internal 🔒 | 🛑  | |
| └ | functionCallWithValue | Internal 🔒 | 🛑  | |
| └ | functionCallWithValue | Internal 🔒 | 🛑  | |
| └ | functionStaticCall | Internal 🔒 |   | |
| └ | functionStaticCall | Internal 🔒 |   | |
| └ | functionDelegateCall | Internal 🔒 | 🛑  | |
| └ | functionDelegateCall | Internal 🔒 | 🛑  | |
| └ | verifyCallResult | Internal 🔒 |   | |
||||||
| **IERC20** | Interface |  |||
| └ | totalSupply | External ❗ |   |NO❗ |
| └ | balanceOf | External ❗ |   |NO❗ |
| └ | transfer | External ❗ | 🛑  |NO❗ |
| └ | allowance | External ❗ |   |NO❗ |
| └ | approve | External ❗ | 🛑  |NO❗ |
| └ | transferFrom | External ❗ | 🛑  |NO❗ |
||||||
| **IERC777Sender** | Interface |  |||
| └ | tokensToSend | External ❗ | 🛑  |NO❗ |
||||||
| **IERC777** | Interface |  |||
| └ | name | External ❗ |   |NO❗ |
| └ | symbol | External ❗ |   |NO❗ |
| └ | granularity | External ❗ |   |NO❗ |
| └ | totalSupply | External ❗ |   |NO❗ |
| └ | balanceOf | External ❗ |   |NO❗ |
| └ | send | External ❗ | 🛑  |NO❗ |
| └ | burn | External ❗ | 🛑  |NO❗ |
| └ | isOperatorFor | External ❗ |   |NO❗ |
| └ | authorizeOperator | External ❗ | 🛑  |NO❗ |
| └ | revokeOperator | External ❗ | 🛑  |NO❗ |
| └ | defaultOperators | External ❗ |   |NO❗ |
| └ | operatorSend | External ❗ | 🛑  |NO❗ |
| └ | operatorBurn | External ❗ | 🛑  |NO❗ |
||||||
| **IERC777Recipient** | Interface |  |||
| └ | tokensReceived | External ❗ | 🛑  |NO❗ |
||||||
| **Context** | Implementation |  |||
| └ | _msgSender | Internal 🔒 |   | |
| └ | _msgData | Internal 🔒 |   | |
||||||
| **ERC777** | Implementation | Context, IERC777, IERC20 |||
| └ | <Constructor> | Public ❗ | 🛑  |NO❗ |
| └ | name | Public ❗ |   |NO❗ |
| └ | symbol | Public ❗ |   |NO❗ |
| └ | decimals | Public ❗ |   |NO❗ |
| └ | granularity | Public ❗ |   |NO❗ |
| └ | totalSupply | Public ❗ |   |NO❗ |
| └ | balanceOf | Public ❗ |   |NO❗ |
| └ | send | Public ❗ | 🛑  |NO❗ |
| └ | transfer | Public ❗ | 🛑  |NO❗ |
| └ | burn | Public ❗ | 🛑  |NO❗ |
| └ | isOperatorFor | Public ❗ |   |NO❗ |
| └ | authorizeOperator | Public ❗ | 🛑  |NO❗ |
| └ | revokeOperator | Public ❗ | 🛑  |NO❗ |
| └ | defaultOperators | Public ❗ |   |NO❗ |
| └ | operatorSend | Public ❗ | 🛑  |NO❗ |
| └ | operatorBurn | Public ❗ | 🛑  |NO❗ |
| └ | allowance | Public ❗ |   |NO❗ |
| └ | approve | Public ❗ | 🛑  |NO❗ |
| └ | transferFrom | Public ❗ | 🛑  |NO❗ |
| └ | _mint | Internal 🔒 | 🛑  | |
| └ | _mint | Internal 🔒 | 🛑  | |
| └ | _send | Internal 🔒 | 🛑  | |
| └ | _burn | Internal 🔒 | 🛑  | |
| └ | _move | Private 🔐 | 🛑  | |
| └ | _approve | Internal 🔒 | 🛑  | |
| └ | _callTokensToSend | Private 🔐 | 🛑  | |
| └ | _callTokensReceived | Private 🔐 | 🛑  | |
| └ | _beforeTokenTransfer | Internal 🔒 | 🛑  | |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗ | 🛑  |NO❗ |
| └ | owner | Public ❗ |   |NO❗ |
| └ | renounceOwnership | Public ❗ | 🛑  | onlyOwner |
| └ | transferOwnership | Public ❗ | 🛑  | onlyOwner |
| └ | _setOwner | Private 🔐 | 🛑  | |
||||||
| **SafeMath** | Library |  |||
| └ | tryAdd | Internal 🔒 |   | |
| └ | trySub | Internal 🔒 |   | |
| └ | tryMul | Internal 🔒 |   | |
| └ | tryDiv | Internal 🔒 |   | |
| └ | tryMod | Internal 🔒 |   | |
| └ | add | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | mul | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
```



```
| └ | div | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
||||||
| **ITR** | Implementation | Ownable, ERC777 |||
| └ | <Constructor> | Public ❗ | 🛑  | ERC777 |
| └ | init | Internal 🔒 | 🛑  | |
| └ | getMaxTotalSupply | Public ❗ |   |NO❗ |
| └ | totalRemaining | Public ❗ |   |NO❗ |
| └ | getClaimFraction | Public ❗ |   |NO❗ |
| └ | claim | External ❗ | 🛑  |NO❗ |
| └ | getGrowthDurationsPassed | 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 “secured”.

✓ No mint function.
✓ No volatile code.
✓ Not many high severity issues were found.
✓ Contract Ownership Renounced.

# 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.