

# SMART CONTRACT

---

## Security Audit Report

Project: Honor DAO Protocol  
Website: [honordao.finance](http://honordao.finance)  
Platform: Binance Chain  
Language: Solidity  
Date: June 11th, 2022

# Table of contents

|                                       |    |
|---------------------------------------|----|
| Introduction .....                    | 4  |
| Project Background .....              | 4  |
| Audit Scope .....                     | 4  |
| Claimed Smart Contract Features ..... | 7  |
| Audit Summary .....                   | 11 |
| Technical Quick Stats .....           | 12 |
| Code Quality .....                    | 13 |
| Documentation .....                   | 13 |
| Use of Dependencies .....             | 13 |
| AS-IS overview .....                  | 14 |
| Severity Definitions .....            | 24 |
| Audit Findings .....                  | 25 |
| Conclusion .....                      | 29 |
| Our Methodology .....                 | 30 |
| Disclaimers .....                     | 32 |
| Appendix                              |    |
| • Code Flow Diagram .....             | 33 |
| • Slither Results Log .....           | 56 |
| • Solidity static analysis .....      | 66 |
| • Solhint Linter .....                | 88 |

THIS IS SECURITY AUDIT REPORT DOCUMENT AND WHICH MAY CONTAIN INFORMATION WHICH IS CONFIDENTIAL. WHICH INCLUDES ANY POTENTIAL VULNERABILITIES AND MALICIOUS CODES WHICH CAN BE USED TO EXPLOIT THE SOFTWARE. THIS MUST BE REFERRED INTERNALLY AND ONLY SHOULD BE MADE AVAILABLE TO THE PUBLIC AFTER ISSUES ARE RESOLVED.

# Introduction

EtherAuthority was contracted by Honor DAO to perform the Security audit of the Honor DAO Protocol smart contracts code. The audit has been performed using manual analysis as well as using automated software tools. This report presents all the findings regarding the audit performed on June 11th, 2022.

**The purpose of this audit was to address the following:**

- Ensure that all claimed functions exist and function correctly.
- Identify any security vulnerabilities that may be present in the smart contract.

## Project Background

Honor DAO Protocol is a Defi Program which has functions like mint, swap, OpenTrade, burn, twap, spot, update, mock, info, transfer, set pool, claimable, zap, addLiquidity, cleanDust, etc. The Honor DAO contract inherits the IERC721, ERC20, SafeERC20, Ownable, ReentrancyGuard, Address, IUniswapV2Router02, IUniswapV2Pair, IERC20, Math, SafeMath, Initializable, ERC20Burnable, TransparentUpgradeableProxy standard smart contracts from the OpenZeppelin library. These OpenZeppelin contracts are considered community-audited and time-tested, and hence are not part of the audit scope.

## Audit scope

| Name            | Code Review and Security Analysis Report for Honor DAO Protocol Smart Contracts |
|-----------------|---------------------------------------------------------------------------------|
| Platform        | Binance Chain / Solidity                                                        |
| File 1          | Pool.sol                                                                        |
| File 1 MD5 Hash | DBA2377A307942CC1A258820CE39ECE2                                                |
| File 2          | SwapStrategyPOL.sol                                                             |
| File 2 MD5 Hash | 36814239B61233184D546EE4BAC29A24                                                |
| File 3          | Timelock.sol                                                                    |
| File 3 MD5 Hash | 94F559046B7CB4335EE0F49341A23DA0                                                |

|                         |                                  |
|-------------------------|----------------------------------|
| <b>File 4</b>           | HonorDaoChef.sol                 |
| <b>File 4 MD5 Hash</b>  | 723A4D8023F357BBBE1FAD15543B5CE1 |
| <b>File 5</b>           | HonorDaoStaking.sol              |
| <b>File 5 MD5 Hash</b>  | C558D0CEB8516B8D0506CD4FE3993C47 |
| <b>File 6</b>           | HonorDaoZapMMSwap.sol            |
| <b>File 6 MD5 Hash</b>  | 8640A54BBC0B412D15BA74964A6DE86F |
| <b>File 7</b>           | NFTController.sol                |
| <b>File 7 MD5 Hash</b>  | 7B517FFAE5E28C8D3B7020747FFA8659 |
| <b>File 8</b>           | NFTControllerProxy.sol           |
| <b>File 8 MD5 Hash</b>  | E2E8A433C71CE32907140DCF3F28480D |
| <b>File 9</b>           | Fund.sol                         |
| <b>File 9 MD5 Hash</b>  | 47370A0301A3BBA40747C7FFD8A18E6B |
| <b>File 10</b>          | DaoFund.sol                      |
| <b>File 10 MD5 Hash</b> | 03E88C7F44DA8F457B869D1458D326D2 |
| <b>File 11</b>          | DevFund.sol                      |
| <b>File 11 MD5 Hash</b> | B1D2A46F7B6251C4350579D463E4441E |
| <b>File 12</b>          | HONORReserve.sol                 |
| <b>File 12 MD5 Hash</b> | 585DD279DF053AAA9DDFAB3D54E8A809 |
| <b>File 13</b>          | TreasuryFund.sol                 |
| <b>File 13 MD5 Hash</b> | 79648260D8B2C56BAC5C53D30AD1538E |
| <b>File 14</b>          | MockERC20.sol                    |
| <b>File 14 MD5 Hash</b> | 94278D4A01D92E76EBDE914556B3A6A0 |
| <b>File 15</b>          | MockTreasury.sol                 |
| <b>File 15 MD5 Hash</b> | 92986372AD7DF3224D5001BD06A7E7E2 |
| <b>File 16</b>          | MasterOracle.sol                 |
| <b>File 16 MD5 Hash</b> | 26FFB8A6EB84AABF384A830DB4572C0A |
| <b>File 17</b>          | UniswapPairOracle.sol            |

|                         |                                  |
|-------------------------|----------------------------------|
| <b>File 17 MD5 Hash</b> | 37801A23DE6F4571ADD278A4A062C1D5 |
| <b>File 18</b>          | XToken.sol                       |
| <b>File 18 MD5 Hash</b> | E905290FA8FFB182588943AA4D60EAC6 |
| <b>File 19</b>          | YToken.sol                       |
| <b>File 19 MD5 Hash</b> | FFA9BDAB9AEE9D07DB46CB3A23A34696 |
| <b>File 20</b>          | HONOR.sol                        |
| <b>File 20 MD5 Hash</b> | 2380C9128FD368380F28A873EFA536FC |
| <b>File 21</b>          | KNIGHTTestToken.sol              |
| <b>File 21 MD5 Hash</b> | 423D5935E98AB8ADFDA1244E328504B1 |
| <b>File 22</b>          | XNIGHT.sol                       |
| <b>File 22 MD5 Hash</b> | D093B724D075730EA0AF7AE0ECE88401 |
| <b>File 23</b>          | HonorDaoTreasury.sol             |
| <b>File 23 MD5 Hash</b> | D0C2F71B1DC8FE8F68A55DCDA5BD33FB |
| <b>File 24</b>          | StratRecollateralize.sol         |
| <b>File 24 MD5 Hash</b> | C612D113DFC66D0A846B36AA8CDA5700 |
| <b>File 25</b>          | StratReduceReserveLP.sol         |
| <b>File 25 MD5 Hash</b> | 15F43725DBE630F4EDEFD5F93261AF0B |
| <b>Audit Date</b>       | June 11th,2022                   |

## Claimed Smart Contract Features

| Claimed Feature Detail                                                                                                                                                                                                                                                                                                                                                                                                                                   | Our Observation            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
| <b>File 1 Pool.sol</b> <ul style="list-style-type: none"> <li>• Refresh Cooldown: 1 hour</li> <li>• Ratio StepUp: 0.2%</li> <li>• Ratio StepDown: 0.2%</li> <li>• Price Target: 1 KNIGHT</li> <li>• Price Band: 0.004</li> <li>• Minimum Collateral Ratio: 9,00,000</li> <li>• YToken Slippage: 20%</li> <li>• Redemption Fee: 0.5%</li> <li>• Redemption Fee Maximum: 0.9%</li> <li>• Minting Fee: 0.3%</li> <li>• Minting Fee Maximum: 0.5%</li> </ul> | <b>YES, This is valid.</b> |
| <b>File 2 SwapStrategyPOL.sol</b> <ul style="list-style-type: none"> <li>• Swap Slippage: 20%</li> </ul>                                                                                                                                                                                                                                                                                                                                                 | <b>YES, This is valid.</b> |
| <b>File 3 Timelock.sol</b> <ul style="list-style-type: none"> <li>• Grace Period: 14 Days</li> <li>• Minimum Delay: 12 Hours</li> <li>• Maximum Delay: 30 Days</li> </ul>                                                                                                                                                                                                                                                                                | <b>YES, This is valid.</b> |
| <b>File 4 HonorDaoChef.sol</b> <ul style="list-style-type: none"> <li>• Maximum Imumreward: 10 Tokens Per Second</li> <li>• NFT Boost Rate: 100</li> </ul>                                                                                                                                                                                                                                                                                               | <b>YES, This is valid.</b> |
| <b>File 5 HonorDaoStaking.sol</b> <ul style="list-style-type: none"> <li>• Rewards Duration: 1 week</li> <li>• Lock Duration: 4 weeks</li> <li>• Team Reward Percent: 20%</li> </ul>                                                                                                                                                                                                                                                                     | <b>YES, This is valid.</b> |
| <b>File 6 HonorDaoZapMMSwap.sol</b><br>HonorDaoZap is a ZapperFi's simplified version of                                                                                                                                                                                                                                                                                                                                                                 | <b>YES, This is valid.</b> |

|                                                                                                                                                                                                                 |                            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
| <p>zapper contract which will:</p> <ol style="list-style-type: none"> <li>1. use BNB to swap to target token</li> <li>2. make LP between BNB and target token</li> <li>3. add into HonorDaoChef farm</li> </ol> |                            |
| <p><b>File 7 Fund.sol</b></p> <ul style="list-style-type: none"> <li>• Fund has functions like: allocation, transfer, etc.</li> </ul>                                                                           | <b>YES, This is valid.</b> |
| <p><b>File 8 DaoFund.sol</b></p> <ul style="list-style-type: none"> <li>• Allocation: 10%</li> <li>• Vesting Duration: 2 Years</li> </ul>                                                                       | <b>YES, This is valid.</b> |
| <p><b>File 9 DevFund.sol</b></p> <ul style="list-style-type: none"> <li>• Allocation: 10%</li> <li>• Vesting Duration: 1 Years</li> </ul>                                                                       | <b>YES, This is valid.</b> |
| <p><b>File 10 HONORReserve.sol</b></p> <ul style="list-style-type: none"> <li>• MDSReserve has functions like: initialize, setRewarder, etc.</li> </ul>                                                         | <b>YES, This is valid.</b> |
| <p><b>File 11 TreasuryFund.sol</b></p> <ul style="list-style-type: none"> <li>• Allocation: 10%</li> <li>• Vesting Duration: 1 Years</li> </ul>                                                                 | <b>YES, This is valid.</b> |
| <p><b>File 12 MockERC20.sol</b></p> <ul style="list-style-type: none"> <li>• MockERC20 has functions like: mint, etc.</li> </ul>                                                                                | <b>YES, This is valid.</b> |
| <p><b>File 13 MockTreasury.sol</b></p> <ul style="list-style-type: none"> <li>• MockTreasury has functions like: mock, etc.</li> </ul>                                                                          | <b>YES, This is valid.</b> |
| <p><b>File 14 MasterOracle.sol</b></p> <ul style="list-style-type: none"> <li>• MasterOracle has functions like: getYTokenPrice, getYTokenTWAP, etc.</li> </ul>                                                 | <b>YES, This is valid.</b> |
| <p><b>File 15 UniswapPairOracle.sol</b></p> <ul style="list-style-type: none"> <li>• Period: 60-minute Twap (Time-weighted Average Price)</li> </ul>                                                            | <b>YES, This is valid.</b> |

|                                                                                                                                                                      |                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
| <ul style="list-style-type: none"> <li>• Maximum Period: 48 Hours</li> <li>• Minimum Period: 10 Minutes</li> <li>• Leniency: 12 Hours</li> </ul>                     |                            |
| <b>File 16 XToken.sol</b> <ul style="list-style-type: none"> <li>• XToken has functions like: mint, etc.</li> </ul>                                                  | <b>YES, This is valid.</b> |
| <b>File 17 YToken.sol</b> <ul style="list-style-type: none"> <li>• YToken has functions like: burn, etc.</li> </ul>                                                  | <b>YES, This is valid.</b> |
| <b>File 18 HONOR.sol</b> <ul style="list-style-type: none"> <li>• Total Supply: 7 Million ether</li> </ul>                                                           | <b>YES, This is valid.</b> |
| <b>File 19 KNIGHTTestToken.sol</b> <ul style="list-style-type: none"> <li>• Genesis Supply = 100 will be minted as total supply</li> </ul>                           | <b>YES, This is valid.</b> |
| <b>File 20 HonorDaoTreasury.sol</b> <ul style="list-style-type: none"> <li>• HonorDaoTreasury has functions like: balanceOf, requestFund, etc.</li> </ul>            | <b>YES, This is valid.</b> |
| <b>File 21 StratRecollateralize.sol</b> <ul style="list-style-type: none"> <li>• StratRecollateralize has functions like: recollateralize.</li> </ul>                | <b>YES, This is valid.</b> |
| <b>File 22 StratReduceReserveLP.sol</b> <ul style="list-style-type: none"> <li>• StratReduceReserveLP has functions like: reduceReserve, swap.</li> </ul>            | <b>YES, This is valid.</b> |
| <b>File 23 NFTController.sol</b> <ul style="list-style-type: none"> <li>• NFTController has functions like: setBoostRate, setWhitelist, etc.</li> </ul>              | <b>YES, This is valid.</b> |
| <b>File 24 NFTControllerProxy.sol</b> <ul style="list-style-type: none"> <li>• NFTControllerProxy has access to the TransparentUpgradeableProxy contract.</li> </ul> | <b>YES, This is valid.</b> |

|                                                                                                                                            |                            |
|--------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
| <b>File 25 XNIGHT.sol</b> <ul style="list-style-type: none"><li>• XNIGHT has functions like: OpenTrade, includeToWhitelist, etc.</li></ul> | <b>YES, This is valid.</b> |
|--------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|

## Audit Summary

According to the standard audit assessment, Customer's solidity smart contracts are **"Secured"**. Also, these contracts do contain owner control, which does not make them fully decentralized.



We used various tools like Slither, Solhint and Remix IDE. At the same time this finding is based on critical analysis of the manual audit.

All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the Audit overview section. General overview is presented in AS-IS section and all identified issues can be found in the Audit overview section.

**We found 0 critical, 0 high, 0 medium and 2 low and some very low level issues.**

**Investors Advice:** Technical audit of the smart contract does not guarantee the ethical nature of the project. Any owner controlled functions should be executed by the owner with responsibility. All investors/users are advised to do their due diligence before investing in the project.

## Technical Quick Stats

| Main Category        | Subcategory                                   | Result    |
|----------------------|-----------------------------------------------|-----------|
| Contract Programming | Solidity version not specified                | Passed    |
|                      | Solidity version too old                      | Passed    |
|                      | Integer overflow/underflow                    | Passed    |
|                      | Function input parameters lack of check       | Moderated |
|                      | Function input parameters check bypass        | Passed    |
|                      | Function access control lacks management      | Passed    |
|                      | Critical operation lacks event log            | Moderated |
|                      | Human/contract checks bypass                  | Passed    |
|                      | Random number generation/use vulnerability    | N/A       |
|                      | Fallback function misuse                      | Passed    |
|                      | Race condition                                | Passed    |
|                      | Logical vulnerability                         | Passed    |
|                      | Features claimed                              | Passed    |
|                      | Other programming issues                      | Moderated |
| Code Specification   | Function visibility not explicitly declared   | Passed    |
|                      | Var. storage location not explicitly declared | Passed    |
|                      | Use keywords/functions to be deprecated       | Passed    |
|                      | Unused code                                   | Passed    |
| Gas Optimization     | "Out of Gas" Issue                            | Passed    |
|                      | High consumption 'for/while' loop             | Passed    |
|                      | High consumption 'storage' storage            | Passed    |
|                      | Assert() misuse                               | Passed    |
| Business Risk        | The maximum limit for mintage not set         | Moderated |
|                      | "Short Address" Attack                        | Passed    |
|                      | "Double Spend" Attack                         | Passed    |

**Overall Audit Result: PASSED**

## Code Quality

This audit scope has 25 smart contract files. Smart contracts contain Libraries, Smart contracts, inherits and Interfaces. This is a compact and well written smart contract.

The libraries in the Honor DAO Protocol are part of its logical algorithm. A library is a different type of smart contract that contains reusable code. Once deployed on the blockchain (only once), it is assigned a specific address and its properties / methods can be reused many times by other contracts in the Honor DAO Protocol.

The Honor DAO team has not provided unit test scripts, which would have helped to determine the integrity of the code in an automated way.

Some code parts are not well commented on smart contracts. We suggest using Ethereum's NatSpec style for the commenting.

## Documentation

We were given an Honor DAO Protocol smart contract code in the form of a file. The hash of that code is mentioned above in the table.

As mentioned above, code parts are not well commented. But the logic is straightforward. So it is easy to quickly understand the programming flow as well as complex code logic. Comments are very helpful in understanding the overall architecture of the protocol.

Another source of information was its official website <https://honordao.finance> which provided rich information about the project architecture.

## Use of Dependencies

As per our observation, the libraries are used in this smart contracts infrastructure that are based on well known industry standard open source projects.

Apart from libraries, its functions are used in external smart contract calls.

# AS-IS overview

## Pool.sol

### Functions

| Sl. | Functions                   | Type     | Observation       | Conclusion |
|-----|-----------------------------|----------|-------------------|------------|
| 1   | constructor                 | write    | Passed            | No Issue   |
| 2   | owner                       | read     | Passed            | No Issue   |
| 3   | onlyOwner                   | modifier | Passed            | No Issue   |
| 4   | renounceOwnership           | write    | access only Owner | No Issue   |
| 5   | transferOwnership           | write    | access only Owner | No Issue   |
| 6   | transferOwnership           | internal | Passed            | No Issue   |
| 7   | nonReentrant                | modifier | Passed            | No Issue   |
| 8   | info                        | external | Passed            | No Issue   |
| 9   | usableCollateralBalance     | read     | Passed            | No Issue   |
| 10  | calcMint                    | read     | Passed            | No Issue   |
| 11  | calcRedeem                  | read     | Passed            | No Issue   |
| 12  | calcExcessCollateralBalance | read     | Passed            | No Issue   |
| 13  | refreshCollateralRatio      | write    | Passed            | No Issue   |
| 14  | mint                        | external | Passed            | No Issue   |
| 15  | redeem                      | external | Passed            | No Issue   |
| 16  | collect                     | external | Passed            | No Issue   |
| 17  | recollateralize             | external | Passed            | No Issue   |
| 18  | checkPriceFluctuation       | internal | Passed            | No Issue   |
| 19  | toggle                      | write    | access only Owner | No Issue   |
| 20  | setCollateralRatioOptions   | write    | access only Owner | No Issue   |
| 21  | toggleCollateralRatio       | write    | access only Owner | No Issue   |
| 22  | setFees                     | write    | access only Owner | No Issue   |
| 23  | setMinCollateralRatio       | external | access only Owner | No Issue   |
| 24  | reduceExcessCollateral      | external | access only Owner | No Issue   |
| 25  | setSwapStrategy             | external | access only Owner | No Issue   |
| 26  | setOracle                   | external | access only Owner | No Issue   |
| 27  | setYTokenSlippage           | external | access only Owner | No Issue   |
| 28  | setTreasury                 | external | Passed            | No Issue   |
| 29  | transferToTreasury          | internal | Passed            | No Issue   |

## SwapStrategyPOL.sol

### Functions

| Sl. | Functions         | Type     | Observation       | Conclusion |
|-----|-------------------|----------|-------------------|------------|
| 1   | constructor       | write    | Passed            | No Issue   |
| 2   | owner             | read     | Passed            | No Issue   |
| 3   | onlyOwner         | modifier | Passed            | No Issue   |
| 4   | renounceOwnership | write    | access only Owner | No Issue   |

|    |                       |          |                   |          |
|----|-----------------------|----------|-------------------|----------|
| 5  | transferOwnership     | write    | access only Owner | No Issue |
| 6  | _transferOwnership    | internal | Passed            | No Issue |
| 7  | lpBalance             | read     | Passed            | No Issue |
| 8  | execute               | external | Passed            | No Issue |
| 9  | calculateSwapInAmount | internal | Passed            | No Issue |
| 10 | swap                  | internal | Passed            | No Issue |
| 11 | addLiquidity          | internal | Passed            | No Issue |
| 12 | cleanDust             | external | access only Owner | No Issue |
| 13 | changeSlippage        | external | access only Owner | No Issue |

## Timelock.sol

### Functions

| Sl. | Functions          | Type     | Observation | Conclusion |
|-----|--------------------|----------|-------------|------------|
| 1   | constructor        | write    | Passed      | No Issue   |
| 2   | setDelay           | write    | Passed      | No Issue   |
| 3   | acceptAdmin        | write    | Passed      | No Issue   |
| 4   | setPendingAdmin    | write    | Passed      | No Issue   |
| 5   | queueTransaction   | write    | Passed      | No Issue   |
| 6   | cancelTransaction  | write    | Passed      | No Issue   |
| 7   | executeTransaction | write    | Passed      | No Issue   |
| 8   | getBlockTimestamp  | internal | Passed      | No Issue   |

## HonorDaoChef.sol

### Functions

| Sl. | Functions          | Type     | Observation       | Conclusion |
|-----|--------------------|----------|-------------------|------------|
| 1   | constructor        | write    | Passed            | No Issue   |
| 2   | owner              | read     | Passed            | No Issue   |
| 3   | onlyOwner          | modifier | Passed            | No Issue   |
| 4   | renounceOwnership  | write    | access only Owner | No Issue   |
| 5   | transferOwnership  | write    | access only Owner | No Issue   |
| 6   | _transferOwnership | internal | Passed            | No Issue   |
| 7   | poolLength         | read     | Passed            | No Issue   |
| 8   | pendingReward      | external | Passed            | No Issue   |
| 9   | updatePool         | write    | Passed            | No Issue   |
| 10  | massUpdatePools    | write    | Passed            | No Issue   |
| 11  | deposit            | write    | Passed            | No Issue   |
| 12  | withdraw           | write    | Passed            | No Issue   |
| 13  | harvest            | write    | Passed            | No Issue   |
| 14  | withdrawAndHarvest | write    | Passed            | No Issue   |
| 15  | emergencyWithdraw  | write    | Passed            | No Issue   |
| 16  | harvestAllRewards  | external | Passed            | No Issue   |
| 17  | checkPoolDuplicate | internal | Passed            | No Issue   |
| 18  | add                | write    | access only Owner | No Issue   |

|    |                    |          |                   |          |
|----|--------------------|----------|-------------------|----------|
| 19 | set                | write    | access only Owner | No Issue |
| 20 | setRewardPerSecond | write    | access only Owner | No Issue |
| 21 | setRewardMinter    | external | Passed            | No Issue |
| 22 | getBoost           | read     | Passed            | No Issue |
| 23 | getSlots           | read     | Passed            | No Issue |
| 24 | getTokenIds        | read     | Passed            | No Issue |
| 25 | depositNFT         | write    | Passed            | No Issue |
| 26 | withdrawNFT        | write    | Passed            | No Issue |
| 27 | setNftController   | write    | access only Owner | No Issue |
| 28 | setNftBoostRate    | write    | access only Owner | No Issue |

## HonorDaoStaking.sol

### Functions

| Sl. | Functions                | Type     | Observation                                                             | Conclusion           |
|-----|--------------------------|----------|-------------------------------------------------------------------------|----------------------|
| 1   | constructor              | write    | Passed                                                                  | No Issue             |
| 2   | owner                    | read     | Passed                                                                  | No Issue             |
| 3   | onlyOwner                | modifier | Passed                                                                  | No Issue             |
| 4   | renounceOwnership        | write    | access only Owner                                                       | No Issue             |
| 5   | transferOwnership        | write    | access only Owner                                                       | No Issue             |
| 6   | _transferOwnership       | internal | Passed                                                                  | No Issue             |
| 7   | nonReentrant             | modifier | Passed                                                                  | No Issue             |
| 8   | addReward                | write    | Function input parameters lack of check                                 | Refer Audit Findings |
| 9   | approveRewardDistributor | external | access only Owner                                                       | No Issue             |
| 10  | _rewardPerToken          | internal | Passed                                                                  | No Issue             |
| 11  | _earned                  | internal | Passed                                                                  | No Issue             |
| 12  | lastTimeRewardApplicable | read     | Passed                                                                  | No Issue             |
| 13  | getRewardForDuration     | external | Passed                                                                  | No Issue             |
| 14  | claimableRewards         | external | Passed                                                                  | No Issue             |
| 15  | totalBalance             | external | Passed                                                                  | No Issue             |
| 16  | unlockedBalance          | external | Passed                                                                  | No Issue             |
| 17  | earnedBalances           | external | Passed                                                                  | No Issue             |
| 18  | withdrawableBalance      | read     | Passed                                                                  | No Issue             |
| 19  | stake                    | external | Passed                                                                  | No Issue             |
| 20  | mint                     | external | Function input parameters lack of check, Division before multiplication | Refer Audit Findings |
| 21  | withdraw                 | write    | Passed                                                                  | No Issue             |
| 22  | getReward                | write    | Passed                                                                  | No Issue             |
| 23  | emergencyWithdraw        | external | Critical operation lacks event log                                      | Refer Audit Findings |
| 24  | _notifyReward            | internal | Passed                                                                  | No Issue             |
| 25  | notifyRewardAmount       | external | Passed                                                                  | No Issue             |
| 26  | recoverERC20             | external | access only Owner                                                       | No Issue             |

|    |                      |          |        |          |
|----|----------------------|----------|--------|----------|
| 27 | setTeamWalletAddress | external | Passed | No Issue |
| 28 | setTeamRewardPercent | external | Passed | No Issue |
| 29 | updateReward         | modifier | Passed | No Issue |
| 30 | rewardPerToken       | external | Passed | No Issue |
| 31 | lockedBalances       | external | Passed | No Issue |
| 32 | withdrawExpiredLocks | external | Passed | No Issue |

## HonorDaoZapMMSwap.sol

### Functions

| Sl. | Functions             | Type     | Observation       | Conclusion |
|-----|-----------------------|----------|-------------------|------------|
| 1   | constructor           | write    | Passed            | No Issue   |
| 2   | owner                 | read     | Passed            | No Issue   |
| 3   | onlyOwner             | modifier | Passed            | No Issue   |
| 4   | renounceOwnership     | write    | access only Owner | No Issue   |
| 5   | transferOwnership     | write    | access only Owner | No Issue   |
| 6   | _transferOwnership    | internal | Passed            | No Issue   |
| 7   | nonReentrant          | modifier | Passed            | No Issue   |
| 8   | zap                   | external | Passed            | No Issue   |
| 9   | swap                  | internal | Passed            | No Issue   |
| 10  | doSwapETH             | internal | Passed            | No Issue   |
| 11  | approveToken          | internal | Passed            | No Issue   |
| 12  | calculateSwapInAmount | internal | Passed            | No Issue   |
| 13  | addZap                | external | access only Owner | No Issue   |
| 14  | removeZap             | external | access only Owner | No Issue   |

## NFTController.sol

### Functions

| Sl. | Functions              | Type     | Observation       | Conclusion |
|-----|------------------------|----------|-------------------|------------|
| 1   | constructor            | write    | Passed            | No Issue   |
| 2   | owner                  | read     | Passed            | No Issue   |
| 3   | onlyOwner              | modifier | Passed            | No Issue   |
| 4   | renounceOwnership      | write    | access only Owner | No Issue   |
| 5   | transferOwnership      | write    | access only Owner | No Issue   |
| 6   | _transferOwnership     | internal | Passed            | No Issue   |
| 7   | initializer            | modifier | Passed            | No Issue   |
| 8   | reinitializer          | modifier | Passed            | No Issue   |
| 9   | onlyInitializing       | modifier | Passed            | No Issue   |
| 10  | _disableInitializers   | internal | Passed            | No Issue   |
| 11  | _setInitializedVersion | write    | Passed            | No Issue   |
| 12  | initialize             | write    | Passed            | No Issue   |
| 13  | getBoostRate           | external | Passed            | No Issue   |
| 14  | setWhitelist           | external | access only Owner | No Issue   |
| 15  | setDefaultBoostRate    | external | access only Owner | No Issue   |

|    |              |          |                   |          |
|----|--------------|----------|-------------------|----------|
| 16 | setBoostRate | external | access only Owner | No Issue |
|----|--------------|----------|-------------------|----------|

## NFTControllerProxy.sol

### Functions

| Sl. | Functions        | Type     | Observation     | Conclusion |
|-----|------------------|----------|-----------------|------------|
| 1   | constructor      | write    | Passed          | No Issue   |
| 2   | ifAdmin          | modifier | Passed          | No Issue   |
| 3   | admin            | external | access if Admin | No Issue   |
| 4   | implementation   | external | access if Admin | No Issue   |
| 5   | changeAdmin      | external | access if Admin | No Issue   |
| 6   | upgradeTo        | external | access if Admin | No Issue   |
| 7   | upgradeToAndCall | external | access if Admin | No Issue   |
| 8   | admin            | internal | Passed          | No Issue   |
| 9   | _beforeFallback  | internal | Passed          | No Issue   |

## Fund.sol

### Functions

| Sl. | Functions              | Type     | Observation       | Conclusion |
|-----|------------------------|----------|-------------------|------------|
| 1   | constructor            | write    | Passed            | No Issue   |
| 2   | owner                  | read     | Passed            | No Issue   |
| 3   | onlyOwner              | modifier | Passed            | No Issue   |
| 4   | renounceOwnership      | write    | access only Owner | No Issue   |
| 5   | transferOwnership      | write    | access only Owner | No Issue   |
| 6   | _transferOwnership     | internal | Passed            | No Issue   |
| 7   | initializer            | modifier | Passed            | No Issue   |
| 8   | reinitializer          | modifier | Passed            | No Issue   |
| 9   | onlyInitializing       | modifier | Passed            | No Issue   |
| 10  | _disableInitializers   | internal | Passed            | No Issue   |
| 11  | _setInitializedVersion | write    | Passed            | No Issue   |
| 12  | initialize             | external | Passed            | No Issue   |
| 13  | allocation             | read     | Passed            | No Issue   |
| 14  | vestingStart           | read     | Passed            | No Issue   |
| 15  | vestingDuration        | read     | Passed            | No Issue   |
| 16  | currentBalance         | read     | Passed            | No Issue   |
| 17  | vestedBalance          | read     | Passed            | No Issue   |
| 18  | claimable              | read     | Passed            | No Issue   |
| 19  | transfer               | external | access only Owner | No Issue   |

## DaoFund.sol

### Functions

| Sl. | Functions | Type | Observation | Conclusion |
|-----|-----------|------|-------------|------------|
|-----|-----------|------|-------------|------------|

|    |                 |          |                   |          |
|----|-----------------|----------|-------------------|----------|
| 1  | constructor     | write    | Passed            | No Issue |
| 2  | initialize      | external | Passed            | No Issue |
| 3  | allocation      | read     | Passed            | No Issue |
| 4  | vestingStart    | read     | Passed            | No Issue |
| 5  | vestingDuration | read     | Passed            | No Issue |
| 6  | currentBalance  | read     | Passed            | No Issue |
| 7  | vestedBalance   | read     | Passed            | No Issue |
| 8  | claimable       | read     | Passed            | No Issue |
| 9  | transfer        | external | access only Owner | No Issue |
| 10 | allocation      | write    | Passed            | No Issue |
| 11 | vestingStart    | write    | Passed            | No Issue |
| 12 | vestingDuration | write    | Passed            | No Issue |

## DevFund.sol

### Functions

| Sl. | Functions       | Type     | Observation       | Conclusion |
|-----|-----------------|----------|-------------------|------------|
| 1   | constructor     | write    | Passed            | No Issue   |
| 2   | initialize      | external | Passed            | No Issue   |
| 3   | allocation      | read     | Passed            | No Issue   |
| 4   | vestingStart    | read     | Passed            | No Issue   |
| 5   | vestingDuration | read     | Passed            | No Issue   |
| 6   | currentBalance  | read     | Passed            | No Issue   |
| 7   | vestedBalance   | read     | Passed            | No Issue   |
| 8   | claimable       | read     | Passed            | No Issue   |
| 9   | transfer        | external | access only Owner | No Issue   |
| 10  | allocation      | write    | Passed            | No Issue   |
| 11  | vestingStart    | write    | Passed            | No Issue   |
| 12  | vestingDuration | write    | Passed            | No Issue   |

## HONORReserve.sol

### Functions

| Sl. | Functions             | Type     | Observation | Conclusion |
|-----|-----------------------|----------|-------------|------------|
| 1   | constructor           | write    | Passed      | No Issue   |
| 2   | initializer           | modifier | Passed      | No Issue   |
| 3   | reinitializer         | modifier | Passed      | No Issue   |
| 4   | onlyInitializing      | modifier | Passed      | No Issue   |
| 5   | disableInitializers   | internal | Passed      | No Issue   |
| 6   | setInitializedVersion | write    | Passed      | No Issue   |
| 7   | initialize            | external | Passed      | No Issue   |
| 8   | setRewarder           | external | Passed      | No Issue   |
| 9   | setPool               | external | Passed      | No Issue   |
| 10  | transfer              | external | Passed      | No Issue   |

## TreasuryFund.sol

### Functions

| Sl. | Functions       | Type     | Observation       | Conclusion |
|-----|-----------------|----------|-------------------|------------|
| 1   | constructor     | write    | Passed            | No Issue   |
| 2   | initialize      | external | Passed            | No Issue   |
| 3   | allocation      | read     | Passed            | No Issue   |
| 4   | vestingStart    | read     | Passed            | No Issue   |
| 5   | vestingDuration | read     | Passed            | No Issue   |
| 6   | currentBalance  | read     | Passed            | No Issue   |
| 7   | vestedBalance   | read     | Passed            | No Issue   |
| 8   | claimable       | read     | Passed            | No Issue   |
| 9   | transfer        | external | access only Owner | No Issue   |
| 10  | allocation      | write    | Passed            | No Issue   |
| 11  | vestingStart    | write    | Passed            | No Issue   |
| 12  | vestingDuration | write    | Passed            | No Issue   |

## MockERC20.sol

### Functions

| Sl. | Functions           | Type     | Observation | Conclusion |
|-----|---------------------|----------|-------------|------------|
| 1   | constructor         | write    | Passed      | No Issue   |
| 2   | name                | read     | Passed      | No Issue   |
| 3   | symbol              | read     | Passed      | No Issue   |
| 4   | decimals            | read     | Passed      | No Issue   |
| 5   | totalSupply         | read     | Passed      | No Issue   |
| 6   | balanceOf           | read     | Passed      | No Issue   |
| 7   | transfer            | write    | Passed      | No Issue   |
| 8   | allowance           | read     | Passed      | No Issue   |
| 9   | approve             | write    | Passed      | No Issue   |
| 10  | transferFrom        | write    | Passed      | No Issue   |
| 11  | increaseAllowance   | write    | Passed      | No Issue   |
| 12  | decreaseAllowance   | write    | Passed      | No Issue   |
| 13  | transfer            | internal | Passed      | No Issue   |
| 14  | mint                | internal | Passed      | No Issue   |
| 15  | burn                | internal | Passed      | No Issue   |
| 16  | approve             | internal | Passed      | No Issue   |
| 17  | spendAllowance      | internal | Passed      | No Issue   |
| 18  | beforeTokenTransfer | internal | Passed      | No Issue   |
| 19  | afterTokenTransfer  | internal | Passed      | No Issue   |
| 20  | mint                | write    | Passed      | No Issue   |
| 21  | decimals            | read     | Passed      | No Issue   |

## MockTreasury.sol

### Functions

This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

Email: [audit@EtherAuthority.io](mailto:audit@EtherAuthority.io)

| Sl. | Functions   | Type  | Observation | Conclusion |
|-----|-------------|-------|-------------|------------|
| 1   | constructor | write | Passed      | No Issue   |
| 2   | mock        | write | Passed      | No Issue   |
| 3   | info        | read  | Passed      | No Issue   |

## MasterOracle.sol

### Functions

| Sl. | Functions          | Type     | Observation       | Conclusion |
|-----|--------------------|----------|-------------------|------------|
| 1   | constructor        | write    | Passed            | No Issue   |
| 2   | owner              | read     | Passed            | No Issue   |
| 3   | onlyOwner          | modifier | Passed            | No Issue   |
| 4   | renounceOwnership  | write    | access only Owner | No Issue   |
| 5   | transferOwnership  | write    | access only Owner | No Issue   |
| 6   | _transferOwnership | internal | Passed            | No Issue   |
| 7   | getXTokenPrice     | write    | Passed            | No Issue   |
| 8   | getYTokenPrice     | write    | Passed            | No Issue   |
| 9   | getXTokenTWAP      | write    | Passed            | No Issue   |
| 10  | getYTokenTWAP      | write    | Passed            | No Issue   |

## UniswapPairOracle.sol

### Functions

| Sl. | Functions               | Type     | Observation       | Conclusion |
|-----|-------------------------|----------|-------------------|------------|
| 1   | constructor             | write    | Passed            | No Issue   |
| 2   | owner                   | read     | Passed            | No Issue   |
| 3   | onlyOwner               | modifier | Passed            | No Issue   |
| 4   | renounceOwnership       | write    | access only Owner | No Issue   |
| 5   | transferOwnership       | write    | access only Owner | No Issue   |
| 6   | _transferOwnership      | internal | Passed            | No Issue   |
| 7   | setPeriod               | external | access only Owner | No Issue   |
| 8   | update                  | external | Passed            | No Issue   |
| 9   | twap                    | external | Passed            | No Issue   |
| 10  | spot                    | external | Passed            | No Issue   |
| 11  | currentBlockTimestamp   | internal | Passed            | No Issue   |
| 12  | currentCumulativePrices | internal | Passed            | No Issue   |

## XToken.sol

### Functions

| Sl. | Functions   | Type  | Observation | Conclusion |
|-----|-------------|-------|-------------|------------|
| 1   | constructor | write | Passed      | No Issue   |
| 2   | burn        | write | Passed      | No Issue   |
| 3   | burnFrom    | write | Passed      | No Issue   |

|   |            |          |                   |                      |
|---|------------|----------|-------------------|----------------------|
| 4 | onlyMinter | modifier | Passed            | No Issue             |
| 5 | setMinter  | external | Passed            | No Issue             |
| 6 | mint       | external | Unlimited Minting | Refer Audit Findings |

## YToken.sol

### Functions

| Sl. | Functions   | Type  | Observation | Conclusion |
|-----|-------------|-------|-------------|------------|
| 1   | constructor | write | Passed      | No Issue   |
| 2   | burn        | write | Passed      | No Issue   |
| 3   | burnFrom    | write | Passed      | No Issue   |

## HONOR.sol

### Functions

| Sl. | Functions            | Type     | Observation        | Conclusion |
|-----|----------------------|----------|--------------------|------------|
| 1   | constructor          | write    | Passed             | No Issue   |
| 2   | onlyMinter           | modifier | Passed             | No Issue   |
| 3   | setMinter            | external | Passed             | No Issue   |
| 4   | mint                 | external | access only Minter | No Issue   |
| 5   | OpenTrade            | external | Passed             | No Issue   |
| 6   | includeToWhitelist   | write    | Passed             | No Issue   |
| 7   | excludeFromWhitelist | write    | Passed             | No Issue   |
| 8   | _transfer            | internal | Passed             | No Issue   |

## KnightTestToken.sol

### Functions

| Sl. | Functions   | Type     | Observation       | Conclusion           |
|-----|-------------|----------|-------------------|----------------------|
| 1   | constructor | write    | Passed            | No Issue             |
| 2   | burn        | write    | Passed            | No Issue             |
| 3   | burnFrom    | write    | Passed            | No Issue             |
| 4   | onlyMinter  | modifier | Passed            | No Issue             |
| 5   | setMinter   | external | Passed            | No Issue             |
| 6   | mint        | external | Unlimited Minting | Refer Audit Findings |

## XNIGHT.sol

### Functions

| Sl. | Functions   | Type  | Observation | Conclusion |
|-----|-------------|-------|-------------|------------|
| 1   | constructor | write | Passed      | No Issue   |

|   |                      |          |        |          |
|---|----------------------|----------|--------|----------|
| 2 | OpenTrade            | external | Passed | No Issue |
| 3 | includeToWhitelist   | write    | Passed | No Issue |
| 4 | excludeFromWhitelist | write    | Passed | No Issue |
| 5 | _transfer            | internal | Passed | No Issue |

## HonorDaoTreasury.sol

### Functions

| Sl. | Functions          | Type     | Observation       | Conclusion |
|-----|--------------------|----------|-------------------|------------|
| 1   | constructor        | write    | Passed            | No Issue   |
| 2   | owner              | read     | Passed            | No Issue   |
| 3   | onlyOwner          | modifier | Passed            | No Issue   |
| 4   | renounceOwnership  | write    | access only Owner | No Issue   |
| 5   | transferOwnership  | write    | access only Owner | No Issue   |
| 6   | _transferOwnership | internal | Passed            | No Issue   |
| 7   | balanceOf          | read     | Passed            | No Issue   |
| 8   | requestFund        | external | Passed            | No Issue   |
| 9   | addStrategy        | external | access only Owner | No Issue   |
| 10  | removeStrategy     | external | access only Owner | No Issue   |

## StratRecollateralize.sol

### Functions

| Sl. | Functions          | Type     | Observation       | Conclusion |
|-----|--------------------|----------|-------------------|------------|
| 1   | constructor        | write    | Passed            | No Issue   |
| 2   | recollateralize    | external | access only Owner | No Issue   |
| 3   | owner              | read     | Passed            | No Issue   |
| 4   | onlyOwner          | modifier | Passed            | No Issue   |
| 5   | renounceOwnership  | write    | access only Owner | No Issue   |
| 6   | transferOwnership  | write    | access only Owner | No Issue   |
| 7   | _transferOwnership | internal | Passed            | No Issue   |

## StratReduceReserveLP.sol

### Functions

| Sl. | Functions          | Type     | Observation       | Conclusion |
|-----|--------------------|----------|-------------------|------------|
| 1   | constructor        | write    | Passed            | No Issue   |
| 2   | owner              | read     | Passed            | No Issue   |
| 3   | onlyOwner          | modifier | Passed            | No Issue   |
| 4   | renounceOwnership  | write    | access only Owner | No Issue   |
| 5   | transferOwnership  | write    | access only Owner | No Issue   |
| 6   | _transferOwnership | internal | Passed            | No Issue   |
| 7   | reduceReserve      | external | access only Owner | No Issue   |
| 8   | swap               | internal | Passed            | No Issue   |

## Severity Definitions

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

# Audit Findings

## Critical Severity

No Critical severity vulnerabilities were found.

## High Severity

No High severity vulnerabilities were found.

## Medium

No Medium severity vulnerabilities were found.

## Low

(1) Critical operation lacks event log: - [HonorDaoStaking.sol](#)

Missing event log for: emergencyWithdraw

**Resolution:** Write an event log for listed events.

(2) Function input parameters lack of check: - [HonorDaoStaking.sol](#)

Variable validation is not performed in below functions:

- addReward = \_rewardsToken
- mint = user

**Resolution:** We advise to put validation like integer type variables should be greater than 0 and address type variables should not be address(0).

## Very Low / Informational / Best practices:

(1) Unlimited Minting: - [XToken.sol](#)

Minter can mint unlimited tokens.

**Resolution:** We suggest putting a minting limit.

(2) Division before multiplication: [HonorDaoStaking.sol](#)

```
// Mint new tokens
// Minted tokens receive rewards normally but incur a 50% penalty when
// withdrawn before lockDuration has passed.
function mint(address user, uint256 amount) external updateReward(user) {
    require(minters[msg.sender], "MultiFeeDistribution::mint: Only minters allowed");
    totalSupply = totalSupply.add(amount);
    Balances storage bal = balances[user];
    bal.total = bal.total.add(amount);
    bal.earned = bal.earned.add(amount);
    uint256 unlockTime = block.timestamp.div(rewardsDuration).mul(rewardsDuration).add
    LockedBalance[] storage earnings = userEarnings[user];
```

Solidity being resource constraint language, dividing any amount and then multiplying will cause discrepancy in the outcome. Therefore always multiply the amount first and then divide it.

**Resolution:** Consider ordering multiplication before division.

## Centralization

This smart contract has some functions which can be executed by the Admin (Owner) only. If the admin wallet private key would be compromised, then it would create trouble. Following are Admin functions:

- toggle: Pool owner can Turn on / off minting and redemption.
- setCollateralRatioOptions: Pool owner can configure variables related to Collateral Ratio.
- toggleCollateralRatio: Pool Owner can pause or unpause collateral ratio updates.
- setFees: Pool owners can set the protocol fees.
- setMinCollateralRatio: Pool owner can set the minimum Collateral Ratio.
- reduceExcessCollateral: Pool owner can transfer the excess balance of WETH to FeeReserve.
- setSwapStrategy: Pool owner can set the address of Swapper utils.
- setOracle: Pool owner can set new oracle address.
- setYTokenSlippage: Pool owner can set yTokenSlippage.
- setTreasury: Pool owner can set the address of the Treasury.
- cleanDust: SwapStrategyPOL owner can clean dust.
- changeSlippage: SwapStrategyPOL owner can change slippage value.
- add: HonorDaoChef owner can add a new LP to the pool.
- set: HonorDaoChef owner can update the given pool's reward allocation point and `IRewarder` contract.
- setRewardPerSecond: HonorDaoChef owner can set the reward per second to be distributed.
- setRewardMinter: HonorDaoChef owner can set the address of rewardMinter.
- setNftController: HonorDaoChef owner can set NFT controller address.
- setNftBoostRate: HonorDaoChef owner can set NFT Boost Rate value.
- addReward: HonorDaoStaking owner can add a new reward token to be distributed to stakers.
- approveRewardDistributor: HonorDaoStaking owner can modify approval for an address to call notifyRewardAmount.
- recoverERC20: HonorDaoStaking owner can be added to support recovering LP Rewards from other systems such as BAL to be distributed to holders.

- `setTeamWalletAddress:HonorDaoStaking` owner can set the address of the team wallet.
- `setTeamRewardPercent:HonorDaoStaking` owner can set percent of team reward.
- `addZap: HonorDaoZapMMSSwap` owner can add new zap configuration.
- `removeZap: HonorDaoZapMMSSwap` owner can deactivate a Zap configuration.
- `setWhitelist: NFTController` owner can set whitelist address,
- `setDefaultBoostRate: NFTController` owner can set default boost rate value.
- `setBoostRate: NFTController` owner can set boost rate value.
- `transfer: Fund` owners can transfer tokens.
- `transfer: HONORReserve::transfer` owner can only allow funds to withdraw.
- `setPeriod: UniswapPairOracle` owner can set maximum and minimum period.
- `setMinter: XToken` minter can set minter for XToken.
- `mint: XToken` minter can mint new XToken.
- `OpenTrade: HONOR` owners can trade openly.
- `includeToWhitelist: HONOR` owner can include address to whitelist.
- `excludeFromWhitelist: HONOR` owner can exclude address to whitelist.
- `OpenTrade: XNIGHT` owners can trade openly.
- `includeToWhitelist: XNIGHT` owner can include address to whitelist.
- `excludeFromWhitelist: XNIGHT` owner can exclude address to whitelist.
- `addStrategy: HonorDaoTreasury` owner can add new strategy.
- `removeStrategy: HonorDaoTreasury` owner can remove current strategy.
- `allocateFee:HonorDaoTreasury` owner can allocate protocol's fee to stakers.
- `recollateralize: StratRecollateralize` owner can recollateralize the minting pool.
- `reduceReserve: StratReduceReserveLP` owner can remove liquidity, buy back YToken and burn.

To make the smart contract 100% decentralized, we suggest renouncing ownership in the airdrop smart contract once its function is completed.

# Conclusion

We were given a contract code in the form of files. And we have used all possible tests based on given objects as files. We have not observed any major issues in the smart contracts. **So, the smart contracts are ready for the mainnet deployment.**

Since possible test cases can be unlimited for such smart contracts 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.

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools. Smart Contract's high-level description of functionality was presented in the As-is overview section of the report.

Audit report contains all found security vulnerabilities and other issues in the reviewed code.

Security state of the reviewed contract, based on standard audit procedure scope, is **"Secured"**.

# Our Methodology

We like to work with a transparent process and make our reviews a collaborative effort. The goals of our security audits are to improve the quality of systems we review and aim for sufficient remediation to help protect users. The following is the methodology we use in our security audit process.

## **Manual Code Review:**

In manually reviewing all of the code, we look for any potential issues with code logic, error handling, protocol and header parsing, cryptographic errors, and random number generators. We also watch for areas where more defensive programming could reduce the risk of future mistakes and speed up future audits. Although our primary focus is on the in-scope code, we examine dependency code and behavior when it is relevant to a particular line of investigation.

## **Vulnerability Analysis:**

Our audit techniques included manual code analysis, user interface interaction, and whitebox penetration testing. We look at the project's web site to get a high level understanding of what functionality the software under review provides. We then meet with the developers to gain an appreciation of their vision of the software. We install and use the relevant software, exploring the user interactions and roles. While we do this, we brainstorm threat models and attack surfaces. We read design documentation, review other audit results, search for similar projects, examine source code dependencies, skim open issue tickets, and generally investigate details other than the implementation.

## **Documenting Results:**

We follow a conservative, transparent process for analyzing potential security vulnerabilities and seeing them through successful remediation. Whenever a potential issue is discovered, we immediately create an Issue entry for it in this document, even though we have not yet verified the feasibility and impact of the issue. This process is conservative because we document our suspicions early even if they are later shown to not represent exploitable vulnerabilities. We generally follow a process of first documenting the suspicion with unresolved questions, then confirming the issue through code analysis, live experimentation, or automated tests. Code analysis is the most tentative, and we strive to provide test code, log captures, or screenshots demonstrating our confirmation. After this we analyze the feasibility of an attack in a live system.

## **Suggested Solutions:**

We search for immediate mitigations that live deployments can take, and finally we suggest the requirements for remediation engineering for future releases. The mitigation and remediation recommendations should be scrutinized by the developers and deployment engineers, and successful mitigation and remediation is an ongoing collaborative process after we deliver our report, and before the details are made public.

# Disclaimers

## EtherAuthority.io Disclaimer

EtherAuthority team has analyzed this smart contract in accordance with the best industry practices at the date of this report, in relation to: cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report, (Source Code); the Source Code compilation, deployment and functionality (performing the intended functions).

Due to the fact that the total number of test cases are unlimited, the audit makes no statements or warranties on security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bugfree status or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only. We also suggest conducting a bug bounty program to confirm the high level of security of this smart contract.

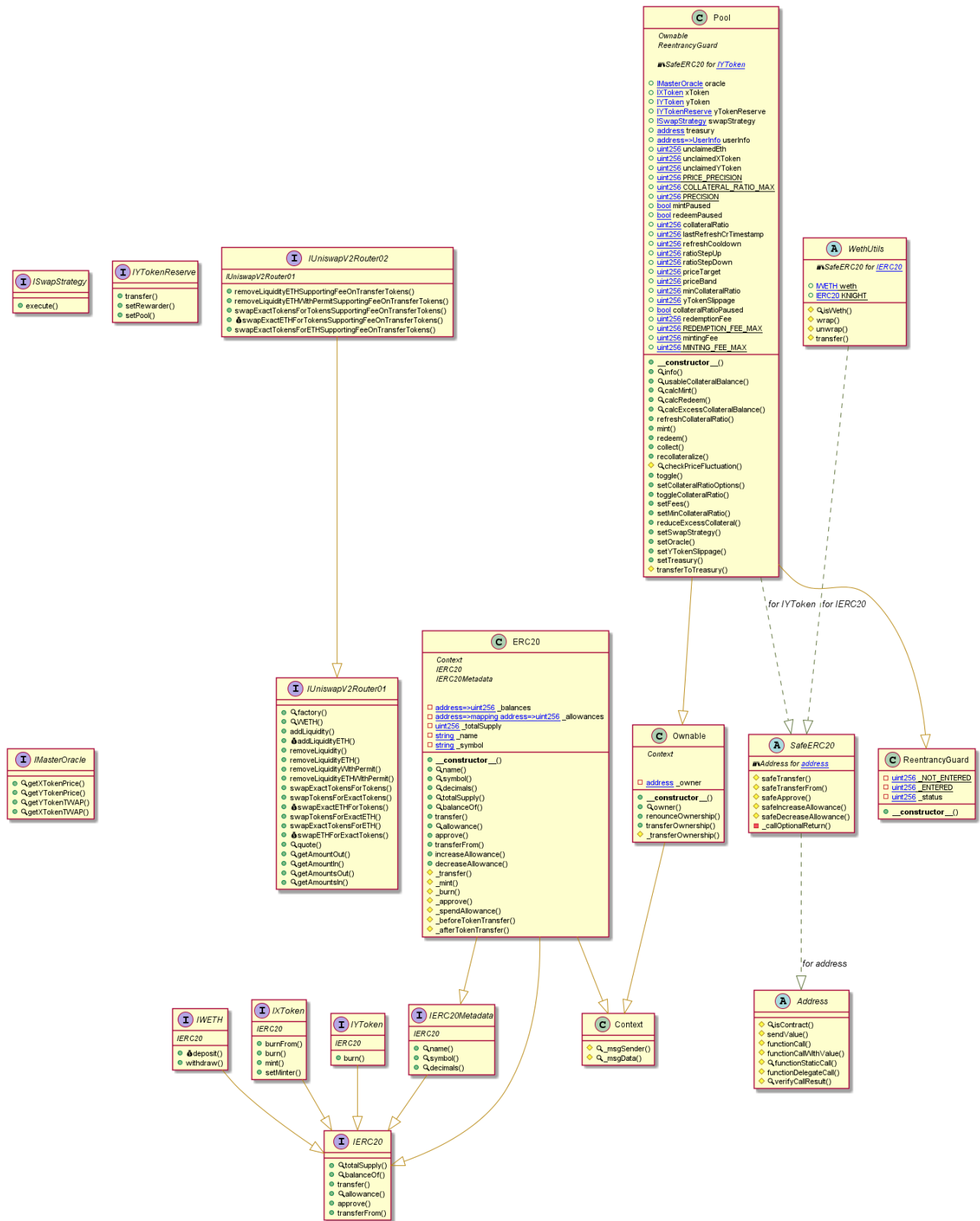
## Technical Disclaimer

Smart contracts are deployed and executed on the blockchain platform. The platform, its programming language, and other software related to the smart contract can have their own vulnerabilities that can lead to hacks. Thus, the audit can't guarantee explicit security of the audited smart contracts.

# Appendix

## Code Flow Diagram - Honor DAO Protocol

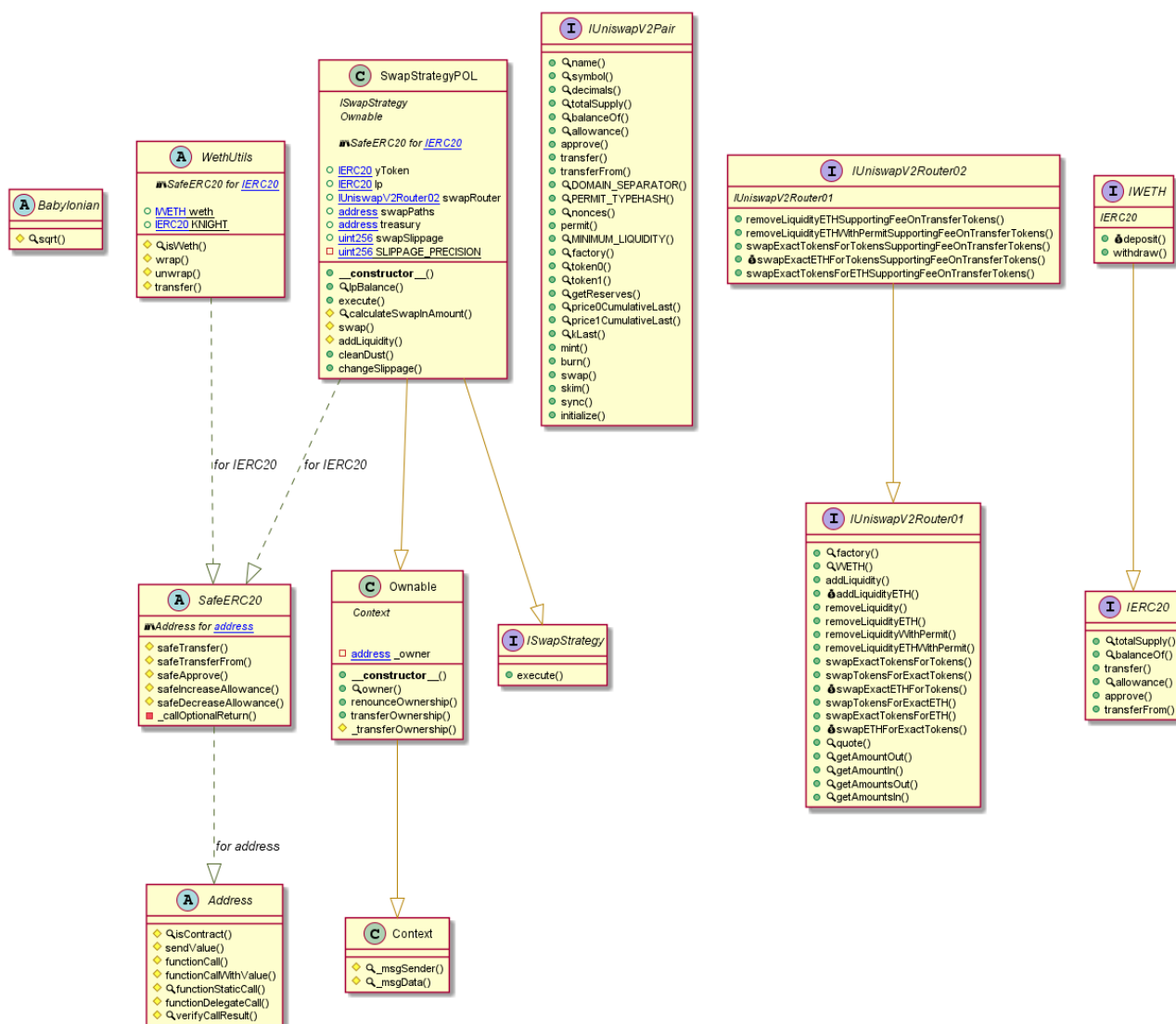
### Pool Diagram



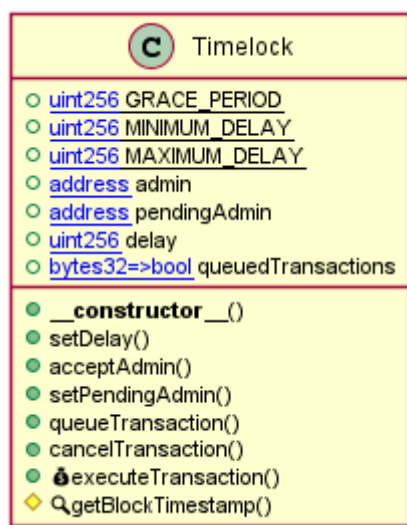
This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

Email: [audit@EtherAuthority.io](mailto:audit@EtherAuthority.io)

## SwapStrategyPOL Diagram



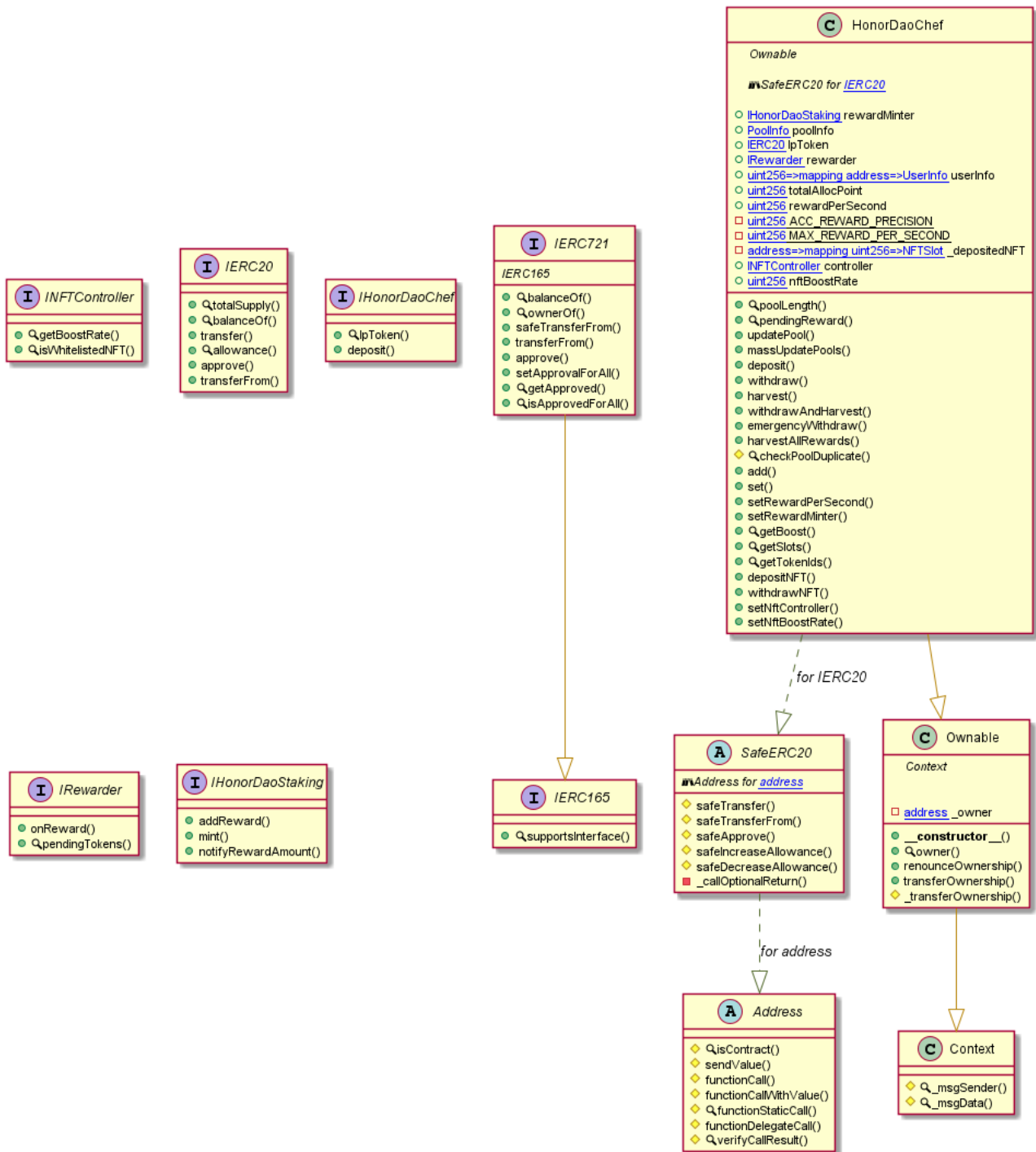
## Timelock Diagram



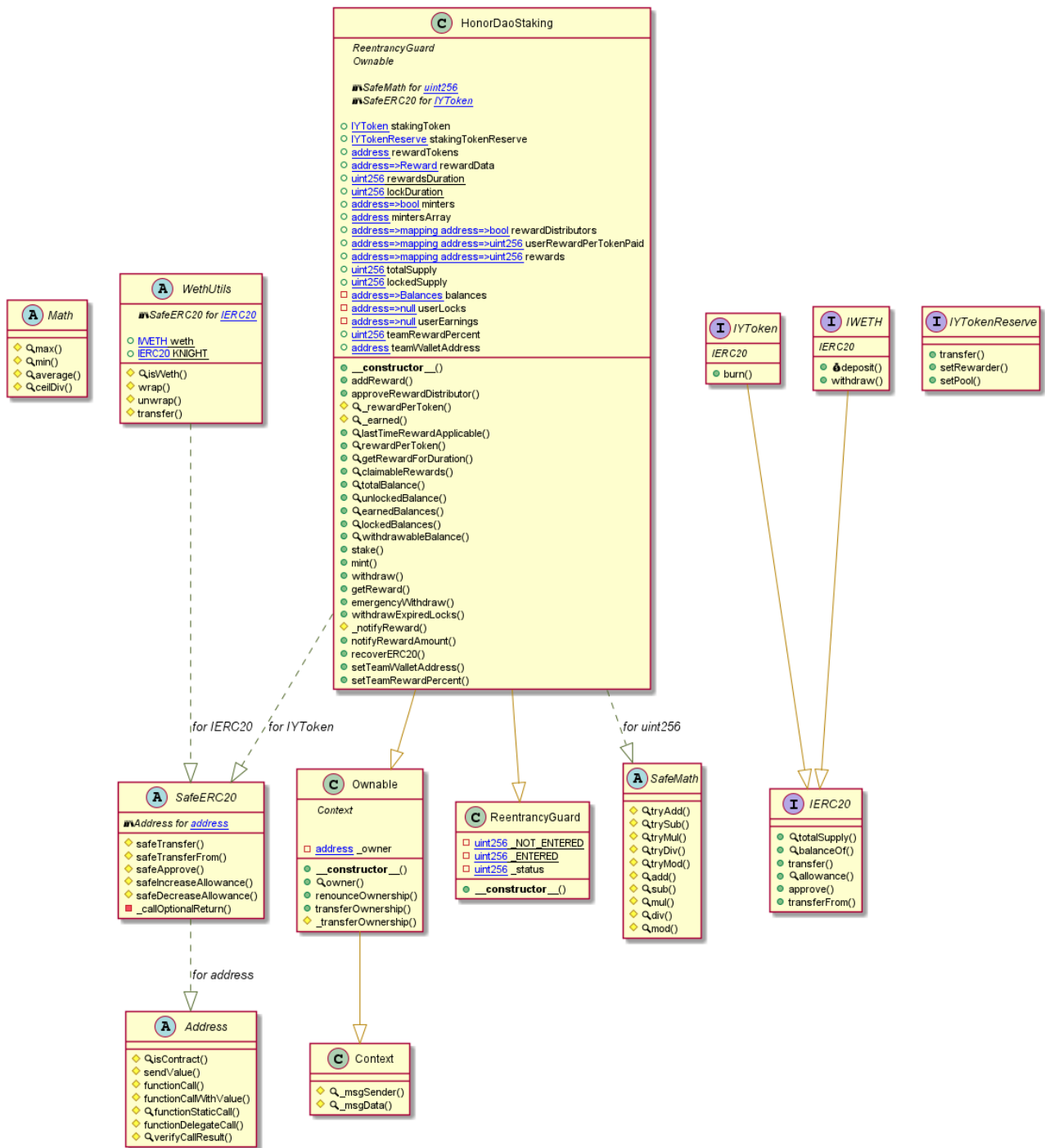
This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

Email: [audit@EtherAuthority.io](mailto:audit@EtherAuthority.io)

# HonorDaoChef Diagram



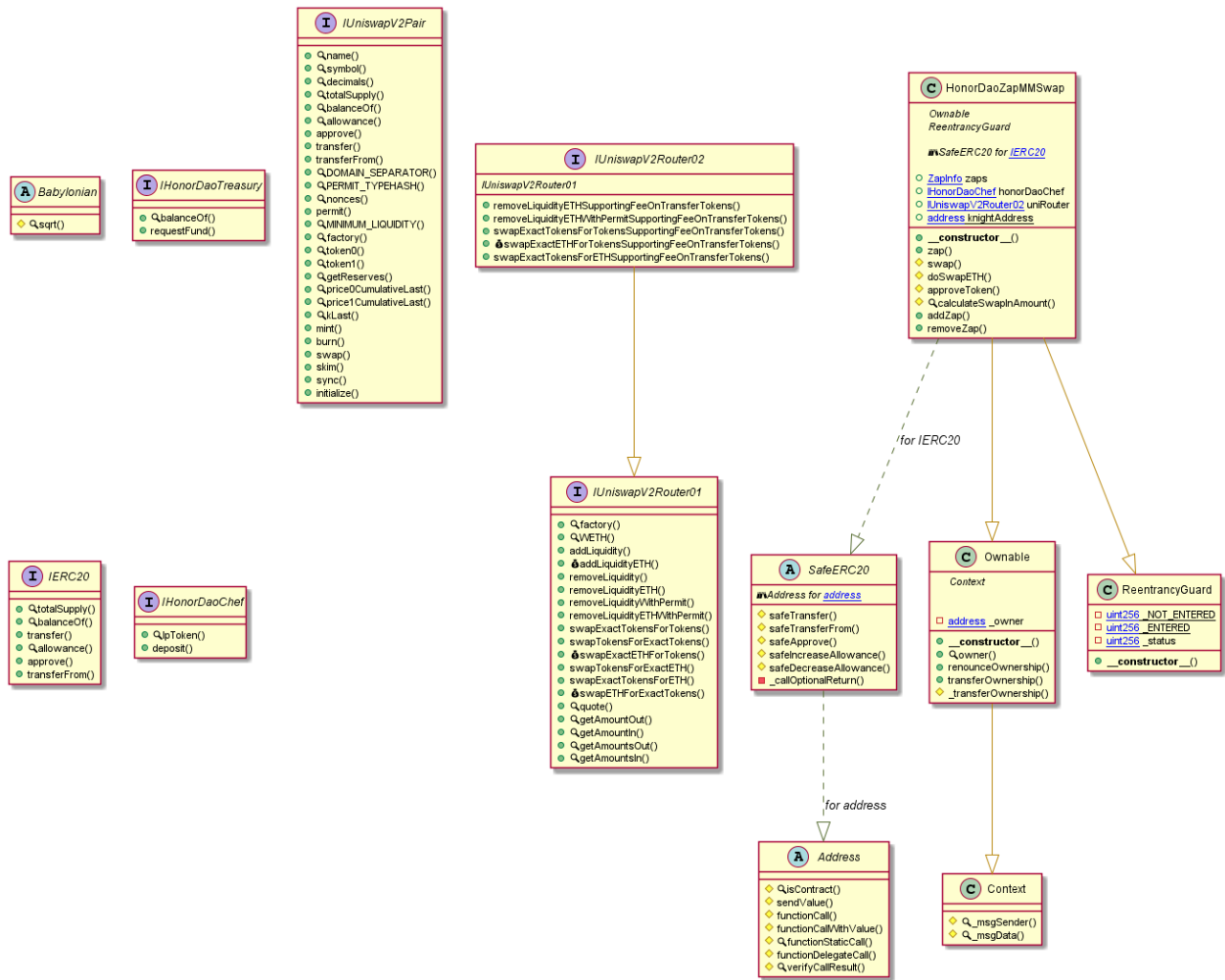
# HonorDaoStaking Diagram



This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

Email: [audit@EtherAuthority.io](mailto:audit@EtherAuthority.io)

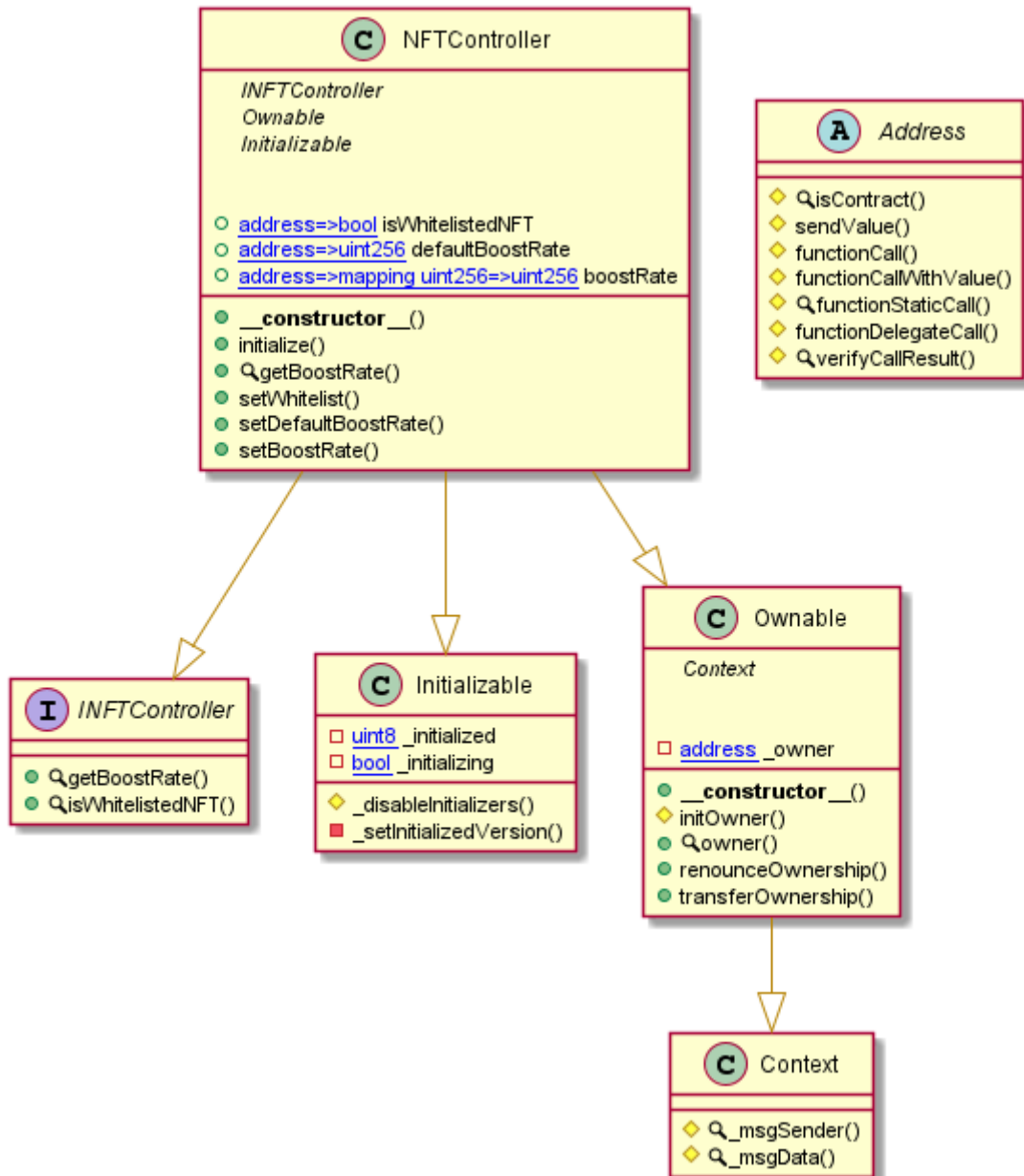
# HonorDaoZapMMSwap Diagram



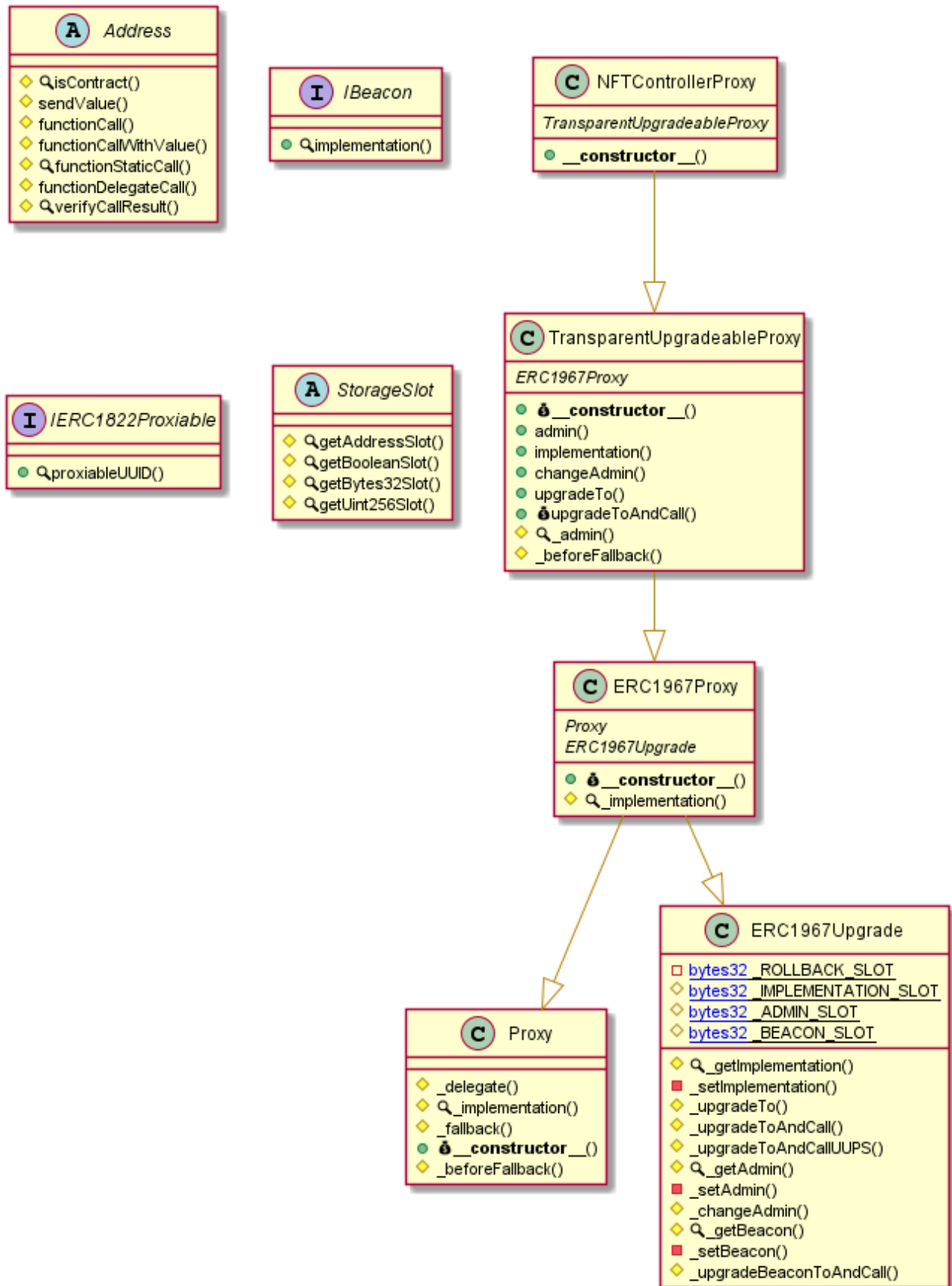
This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

Email: [audit@EtherAuthority.io](mailto:audit@EtherAuthority.io)

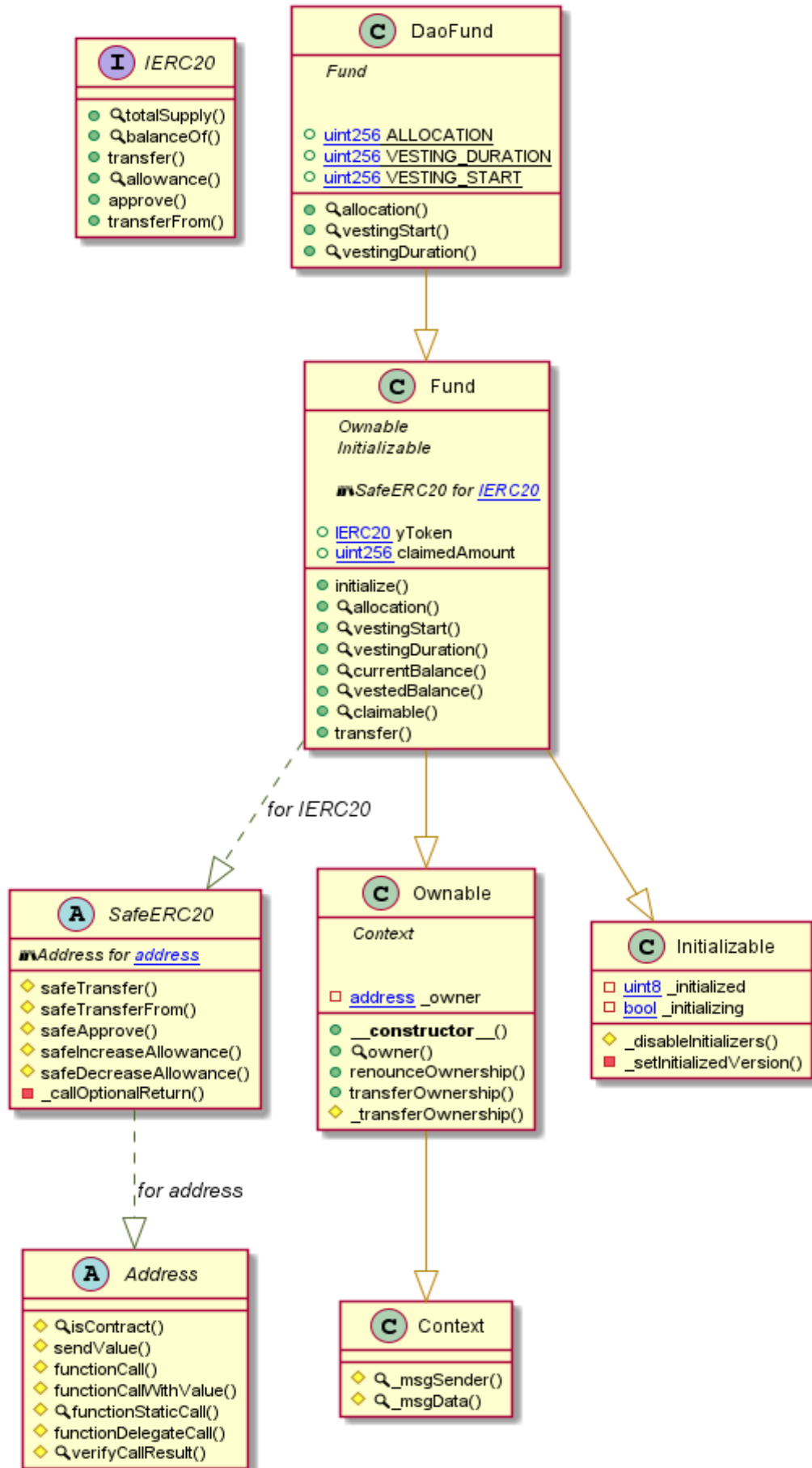
## NFTController Diagram



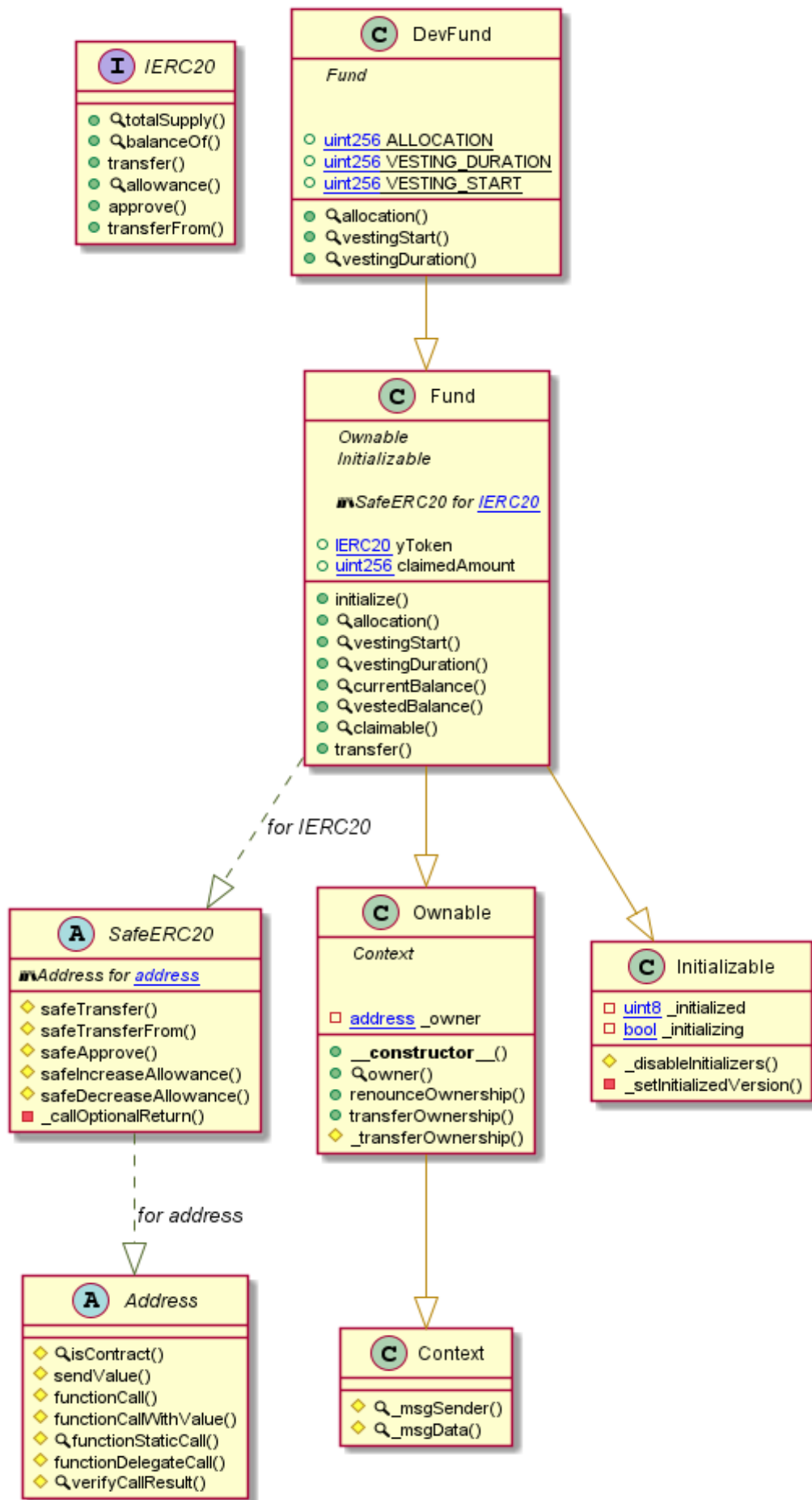
## NFTControllerProxy Diagram



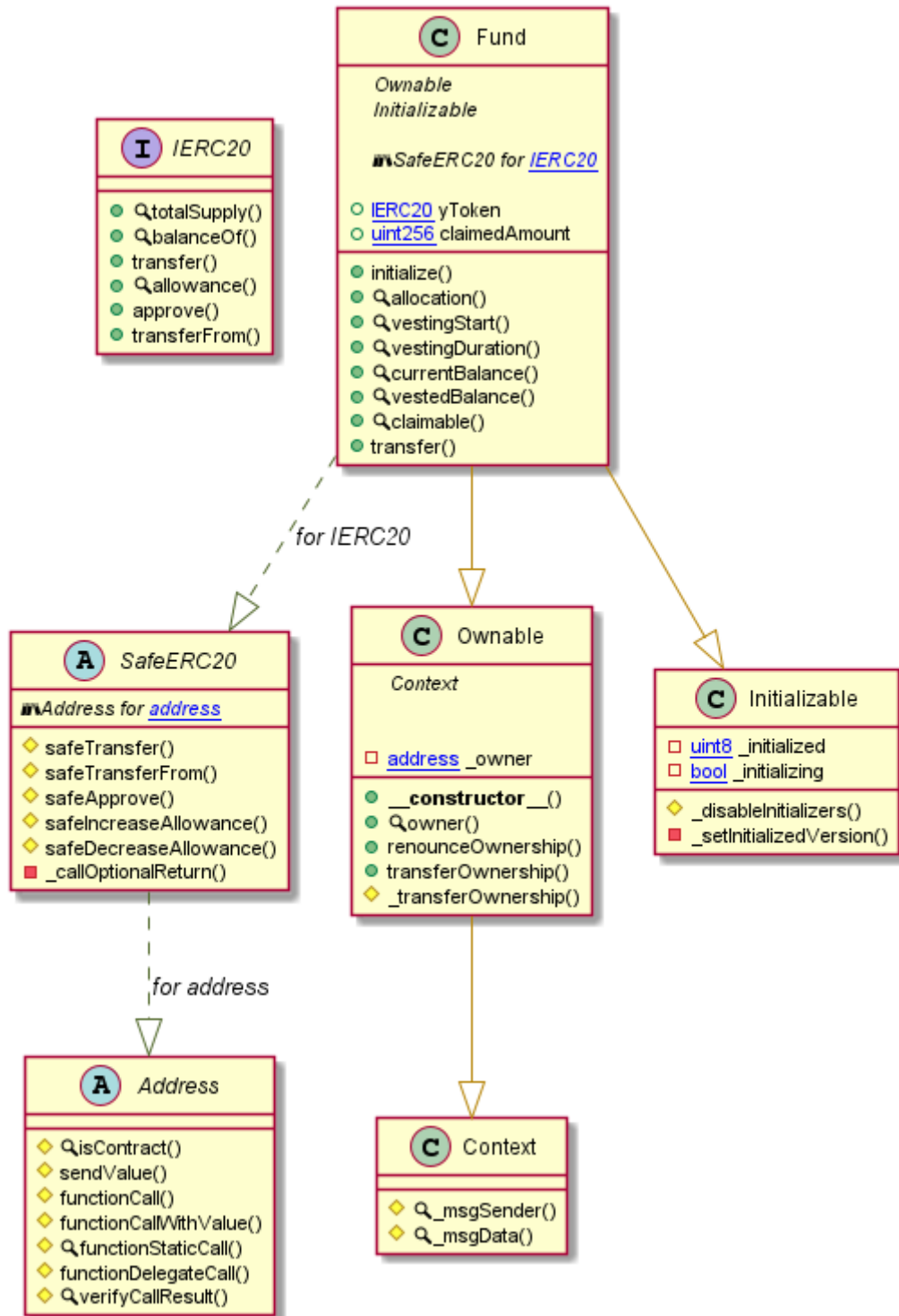
## DaoFund Diagram



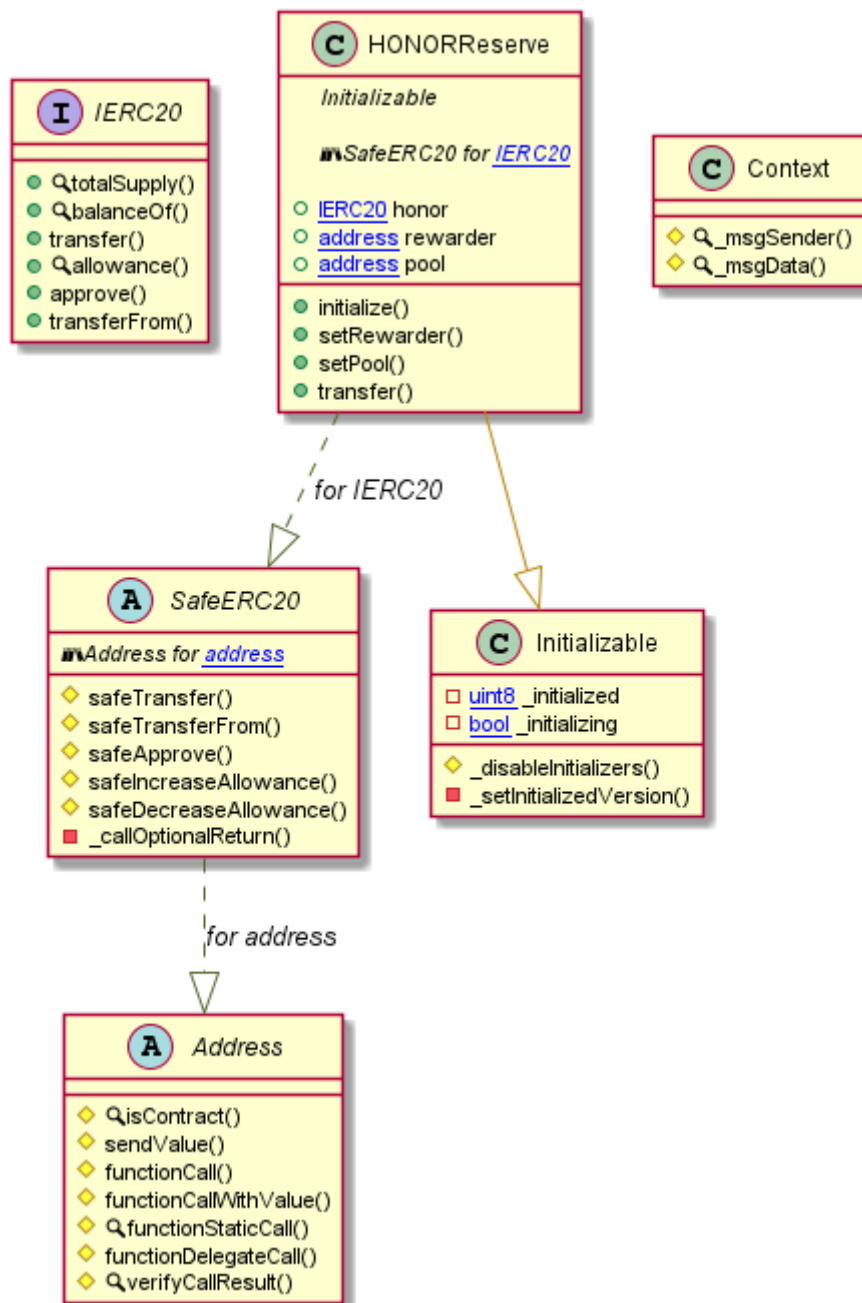
## DevFund Diagram



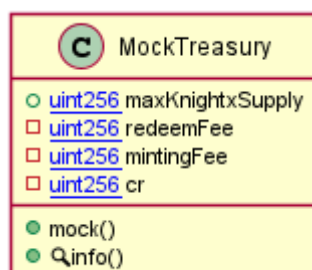
## Fund Diagram



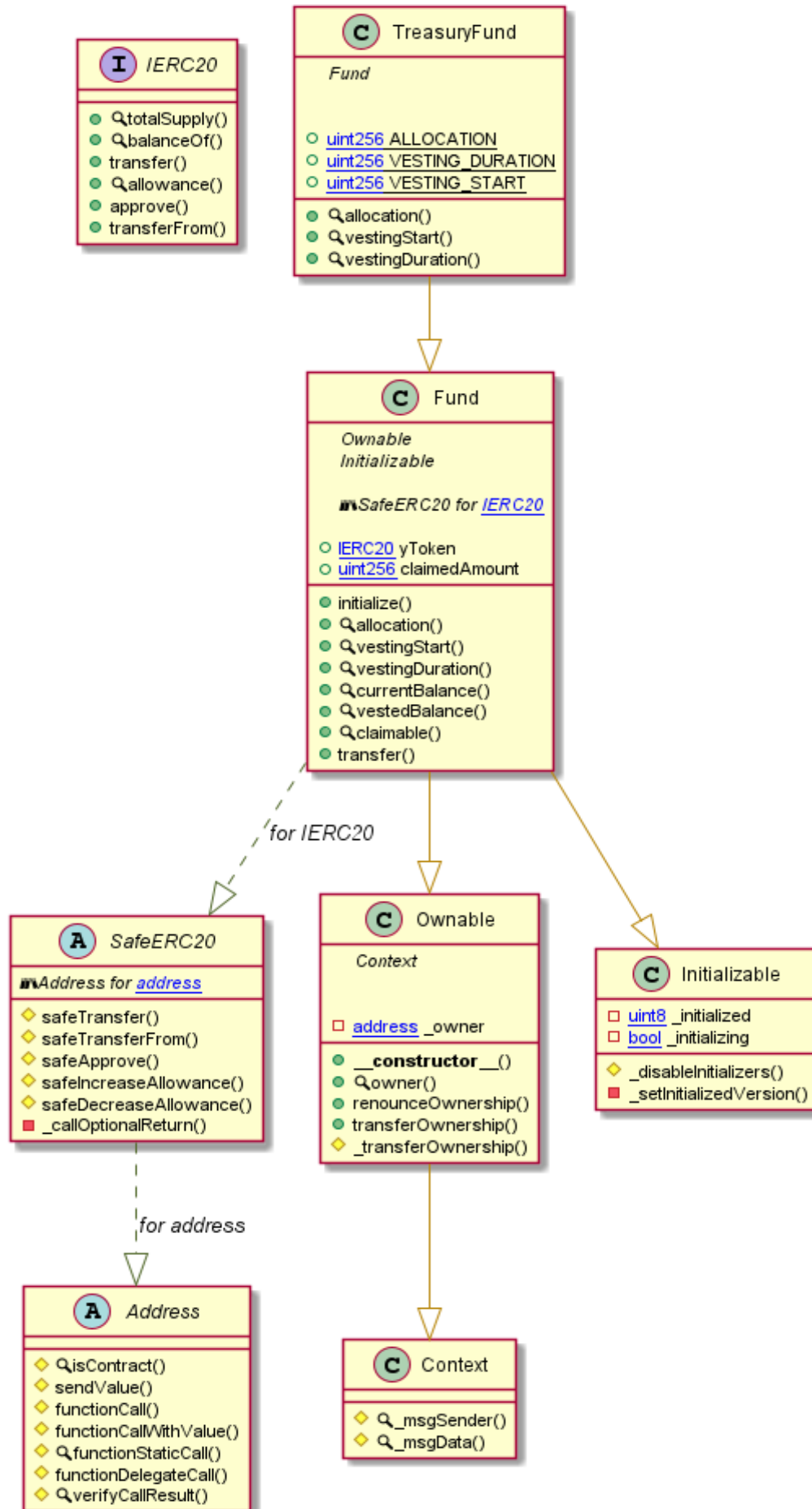
## HONORReserve Diagram



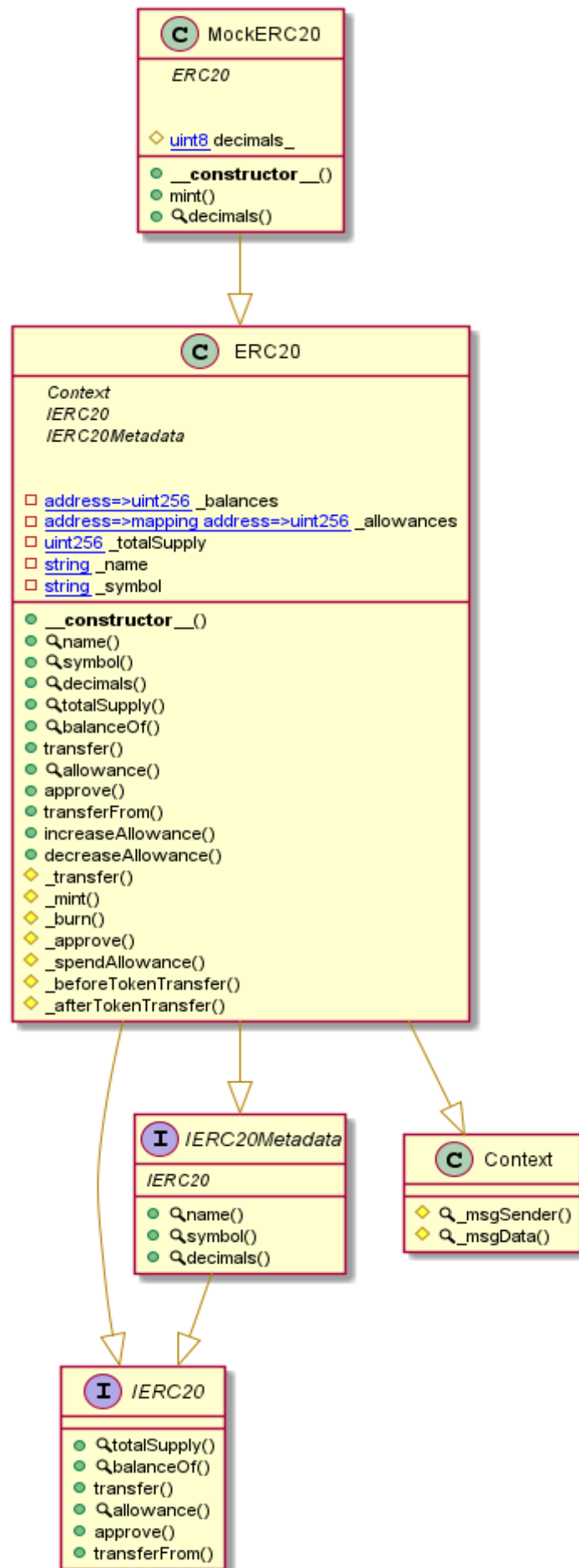
## MockTreasury Diagram



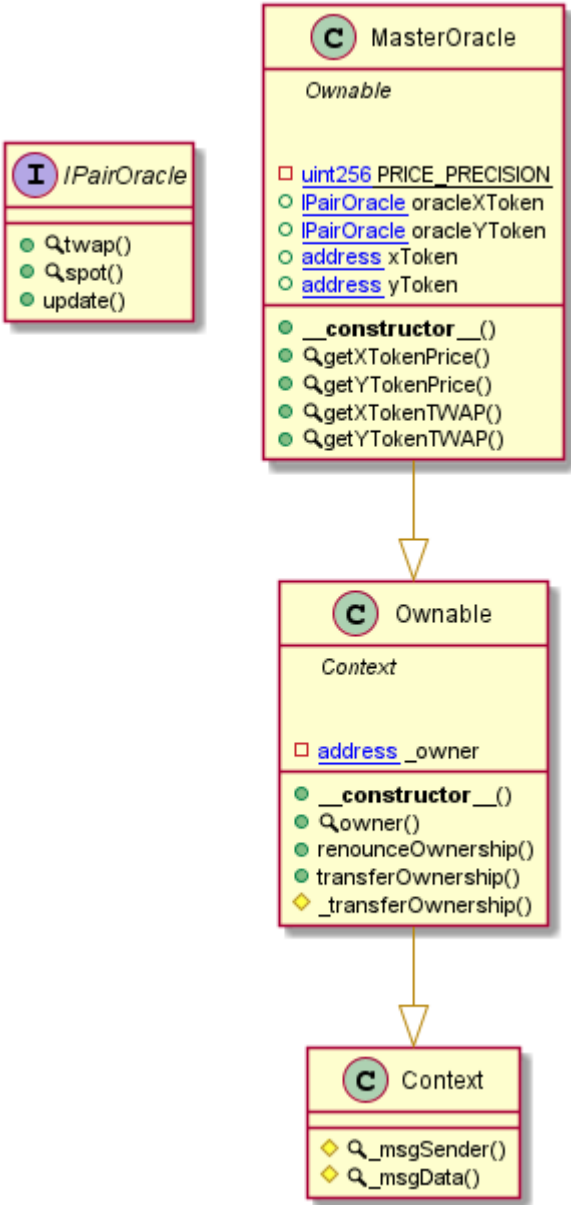
# TreasuryFund Diagram



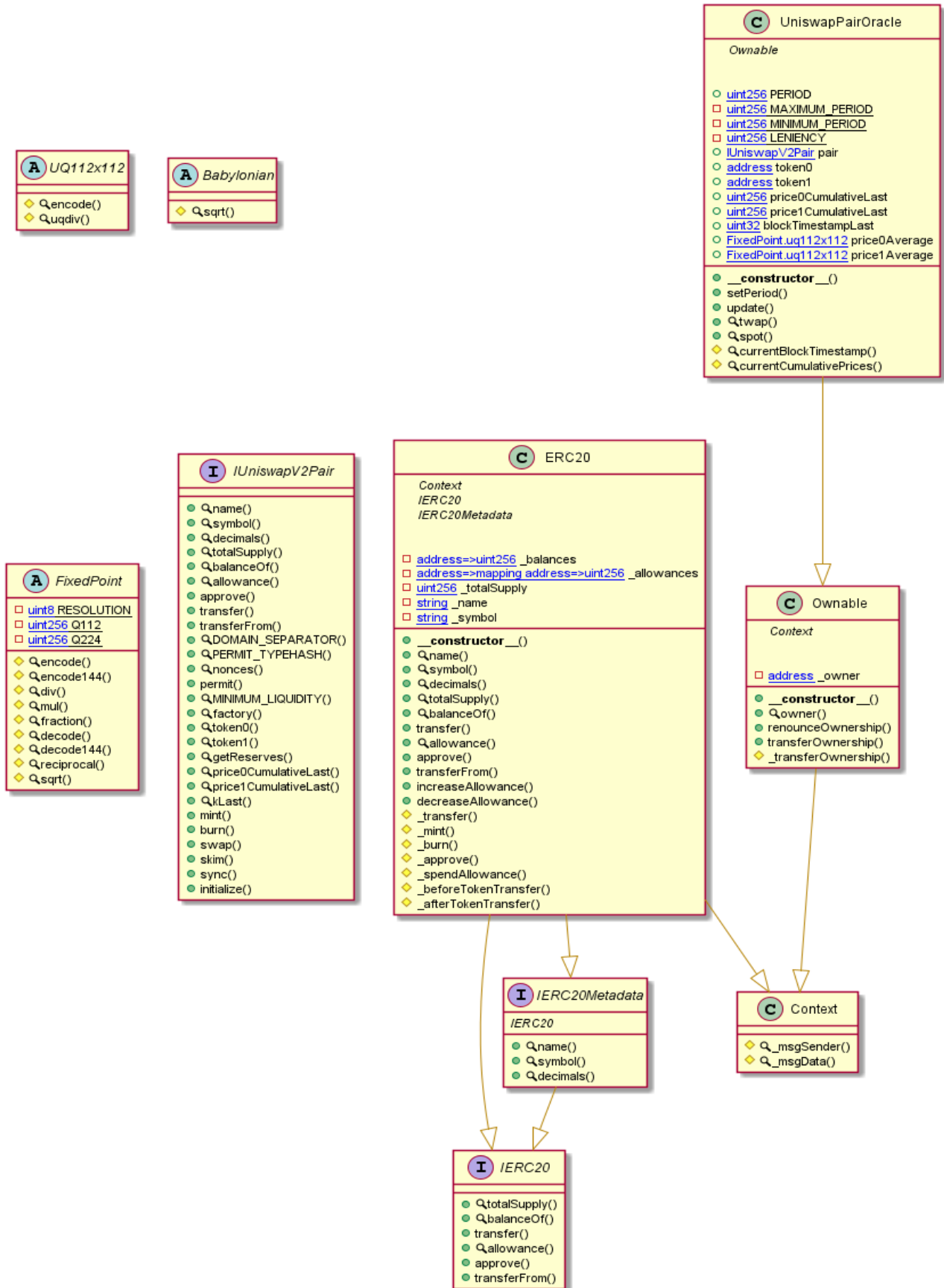
## MockERC20 Diagram



# MasterOracle Diagram



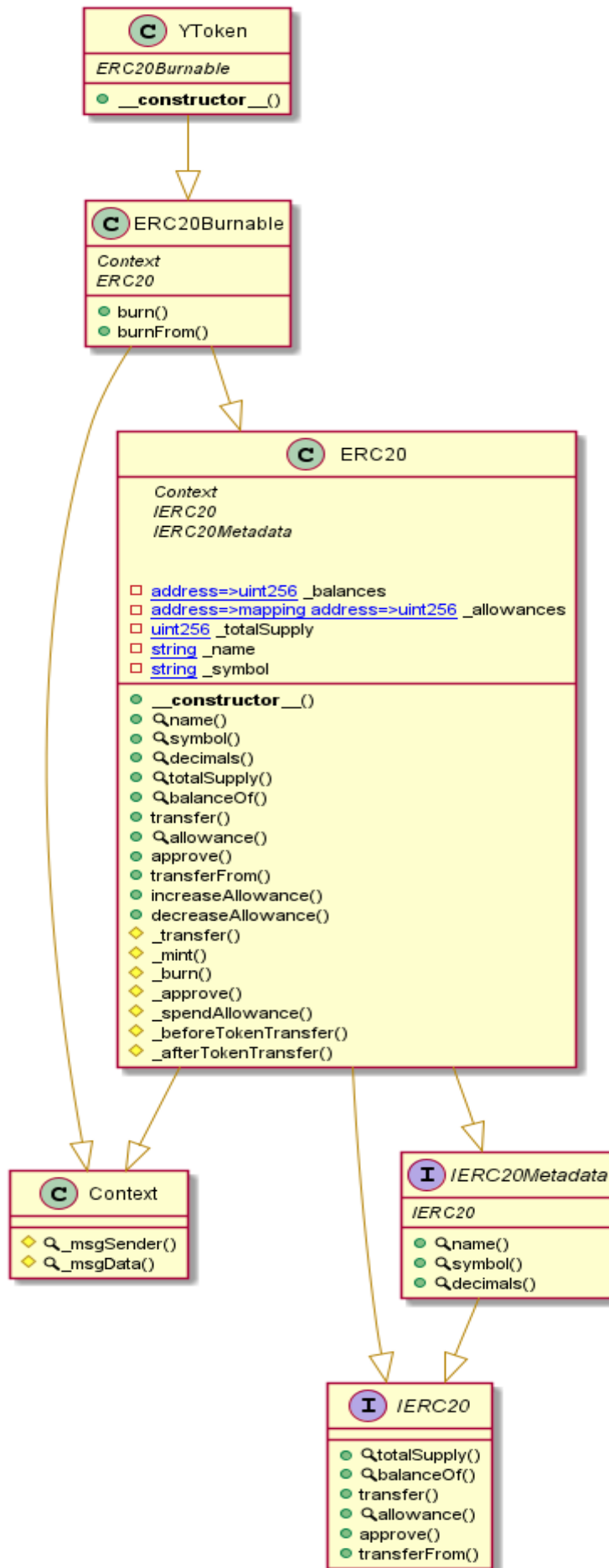
# UniswapPairOracle Diagram



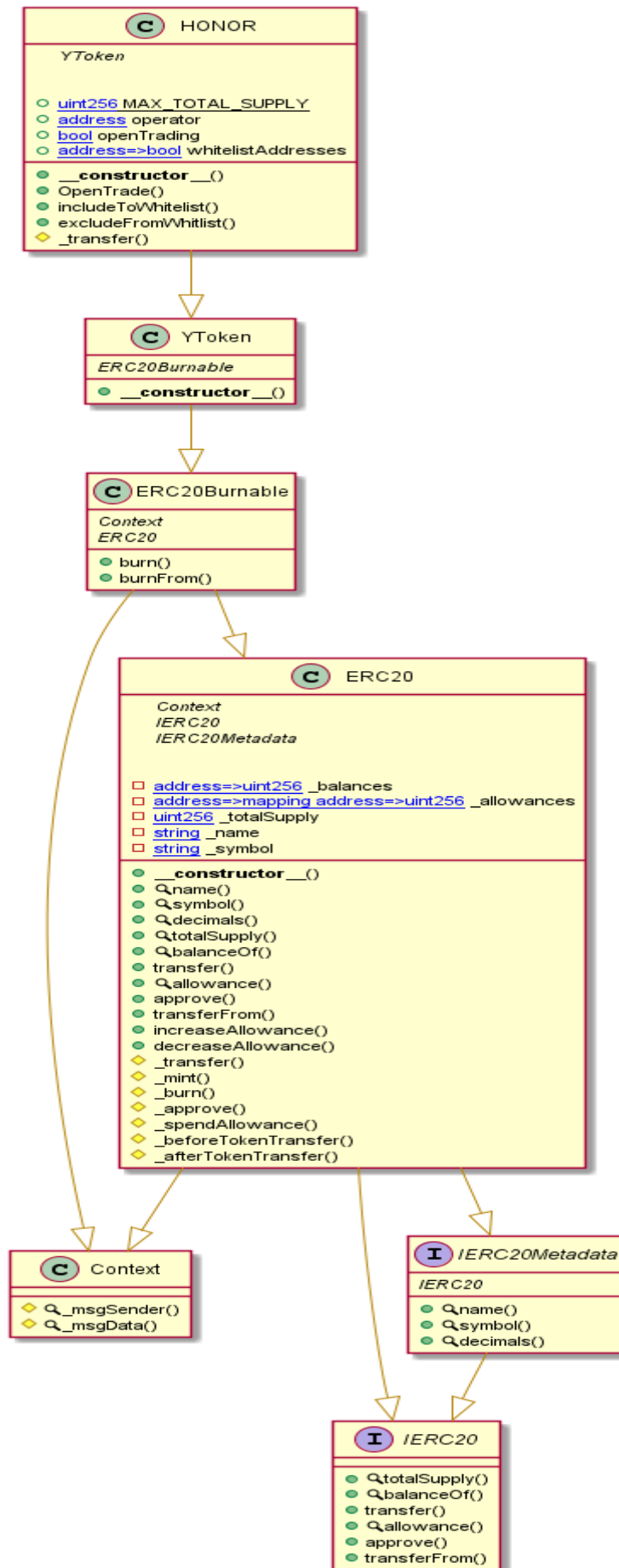
## XToken Diagram



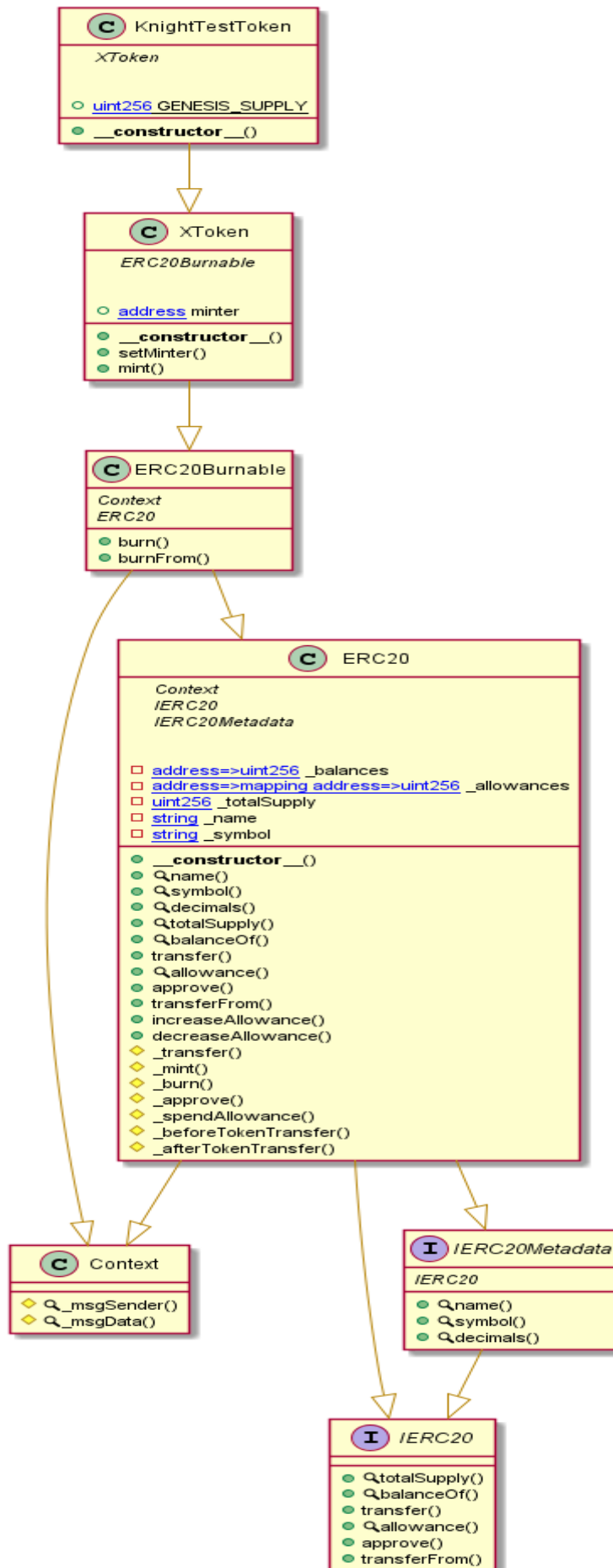
## YToken Diagram



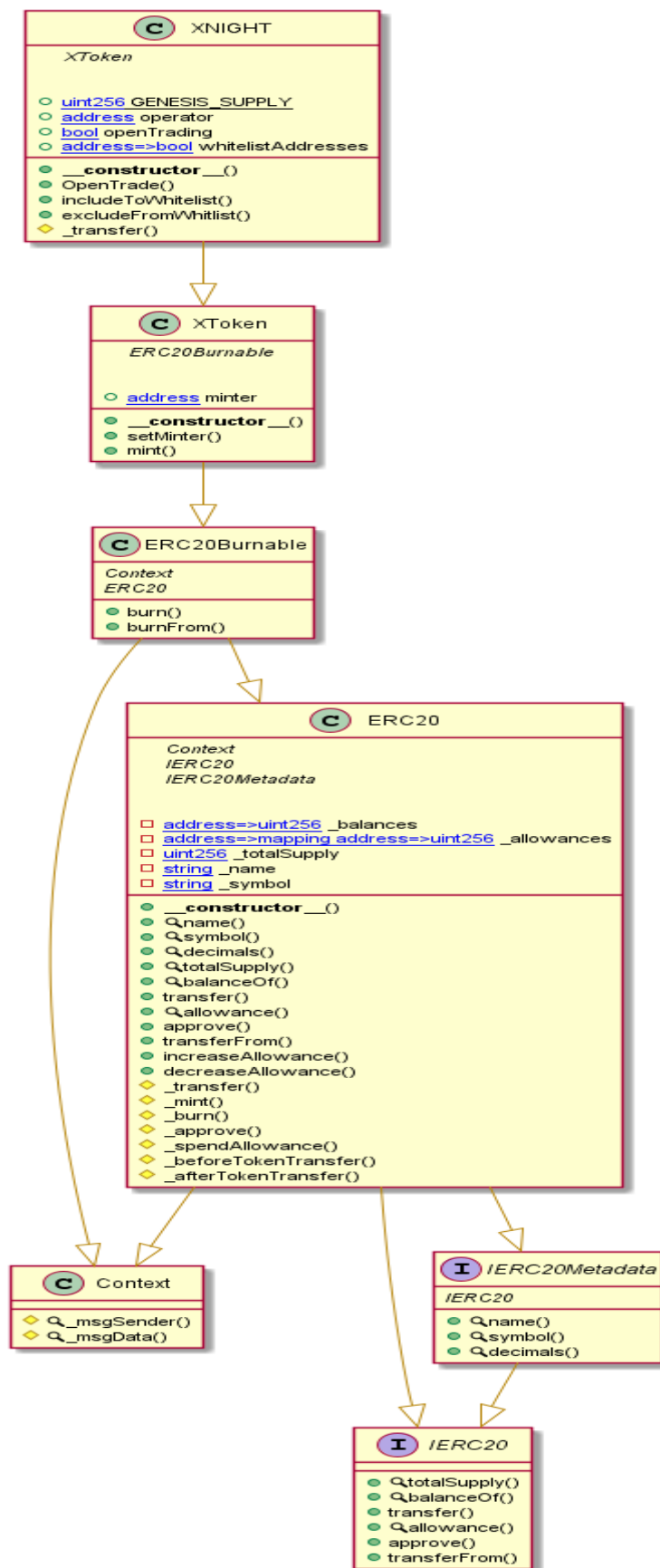
# HONOR Diagram



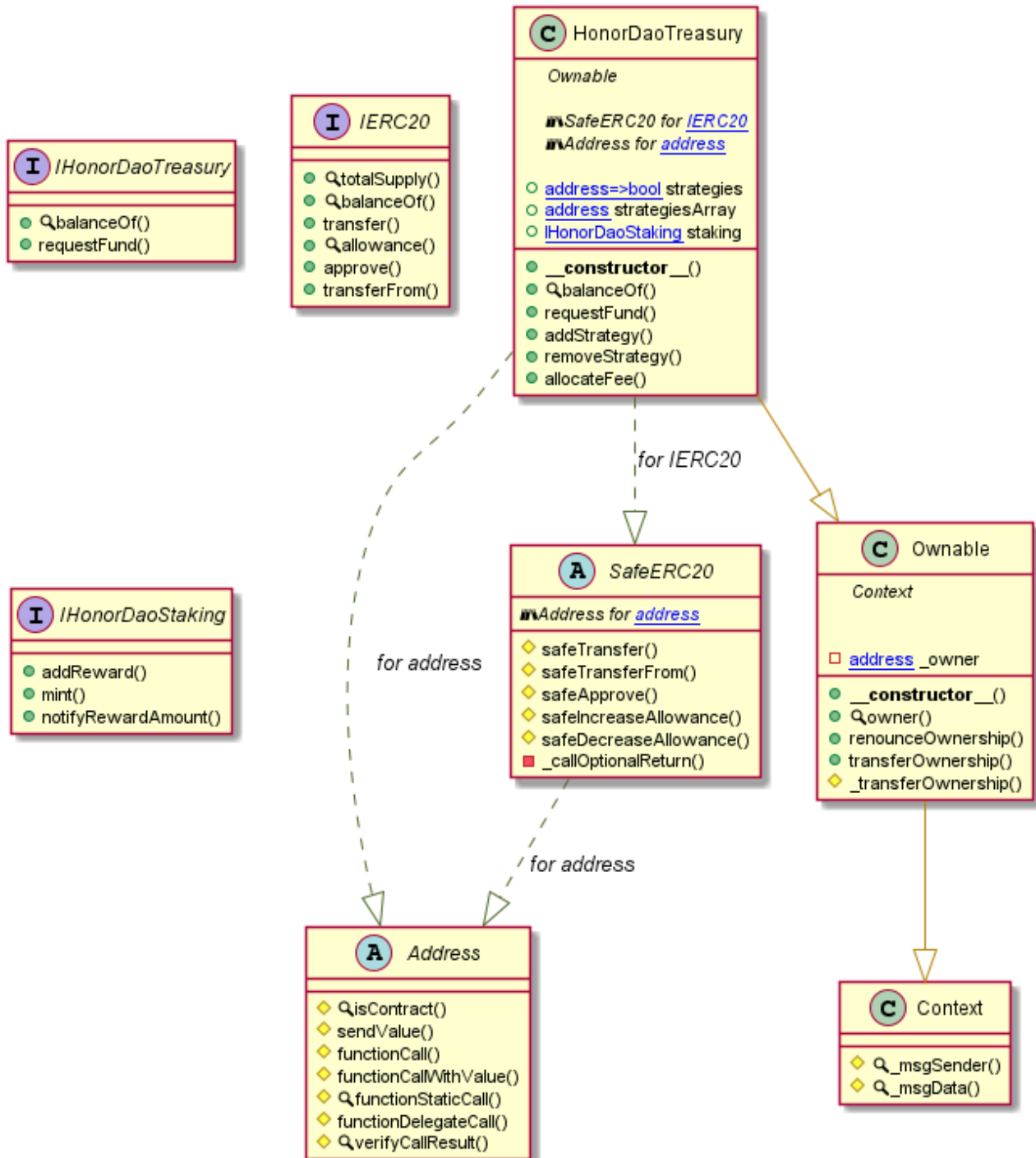
# KNIGHTTestToken Diagram



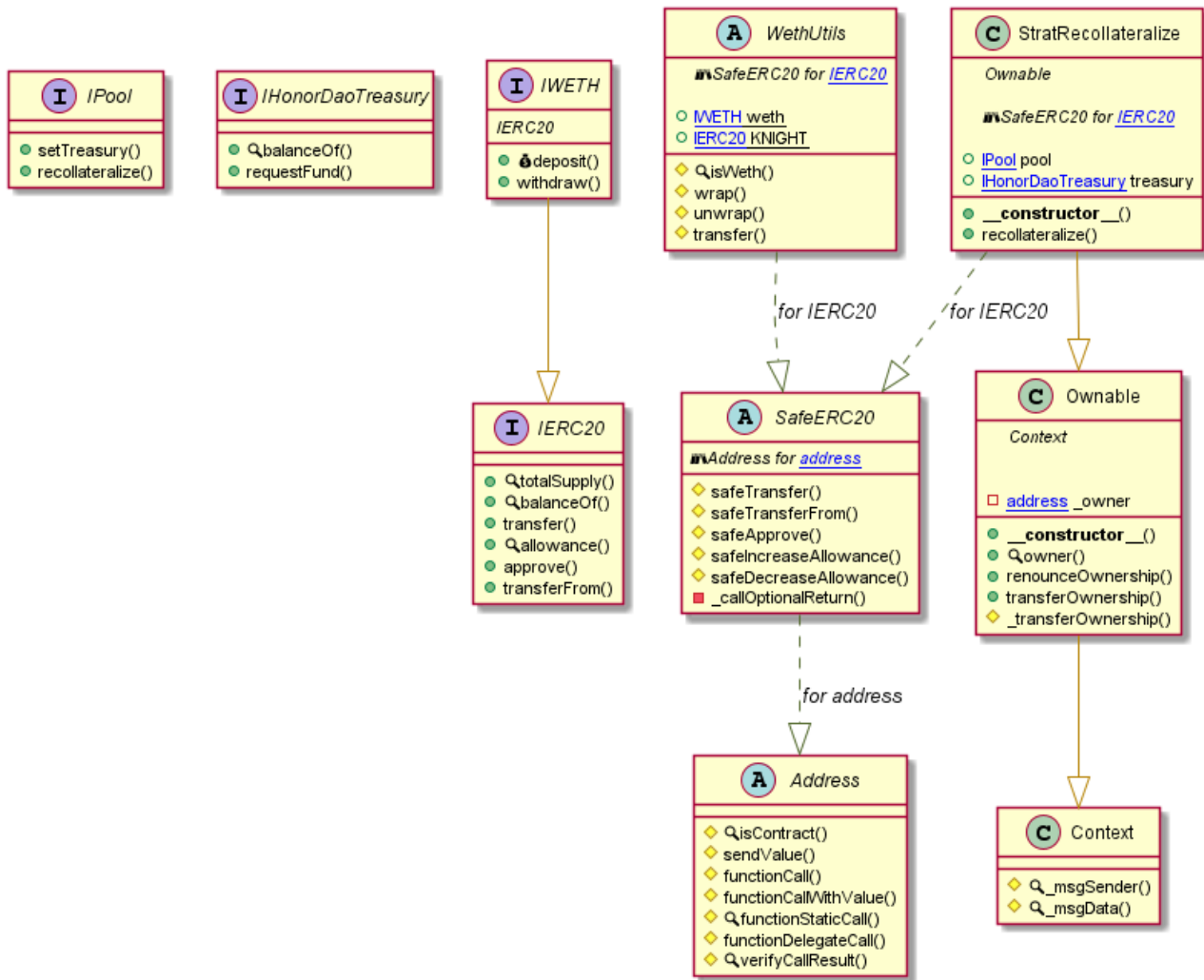
## XNIGHT Diagram



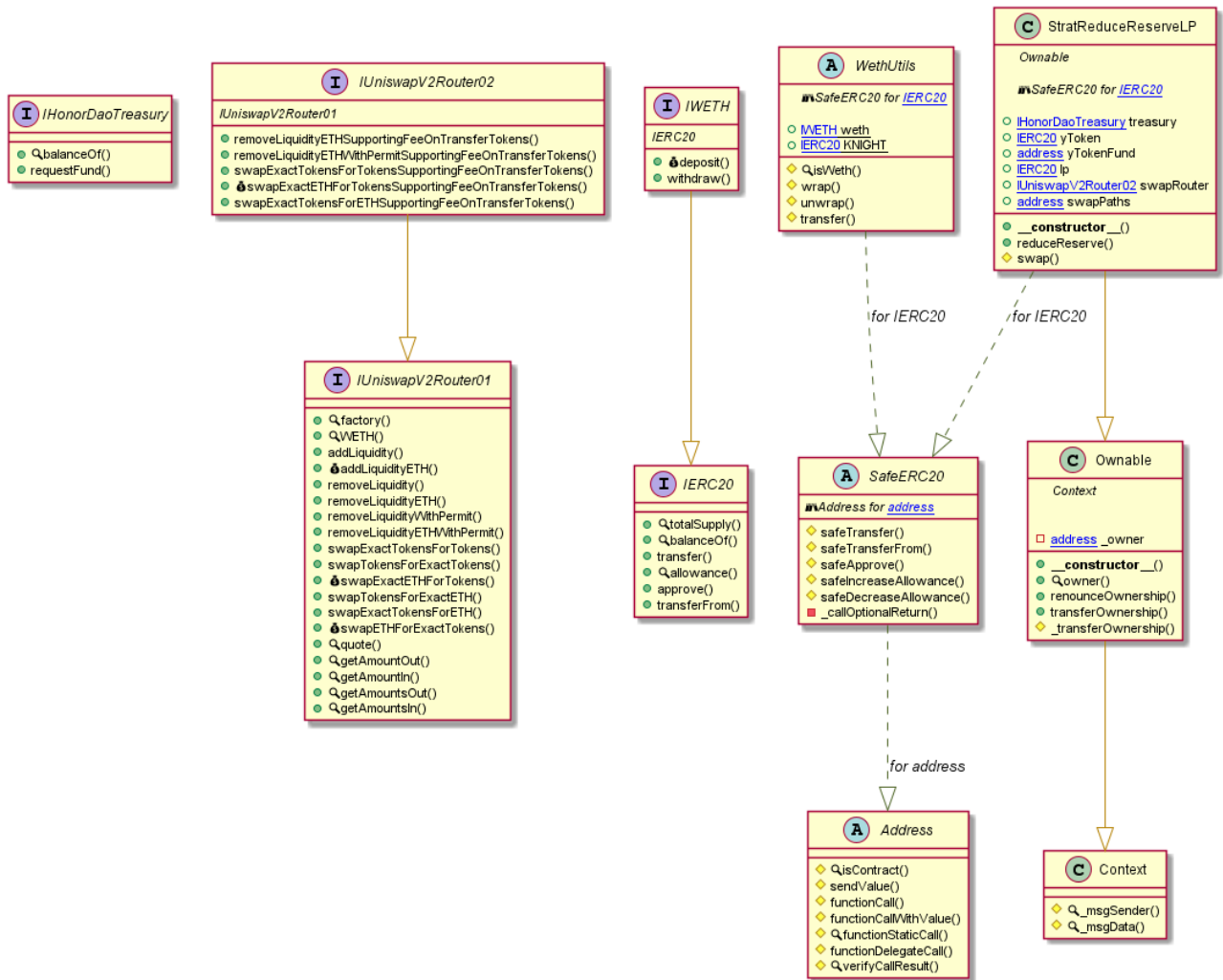
## HonorDaoTreasury Diagram



## StratRecollateralize Diagram



# StratReduceReserveLP Diagram



This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

Email: [audit@EtherAuthority.io](mailto:audit@EtherAuthority.io)

# Slither Results Log

## Slither log >> Pool.sol

```
INFO:Detectors:
Pool.setTreasury(address)._treasury (Pool.sol#1483) lacks a zero-check on :
- treasury = _treasury (Pool.sol#1485)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Reentrancy in Pool.constructor(address,address,address) (Pool.sol#1109-1122):
  External calls:
  - xToken.setMinter(address(this)) (Pool.sol#1118)
  State variables written after the call(s):
  - yToken = IYToken(_yToken) (Pool.sol#1119)
  - yTokenReserve = IYTokenReserve(_yTokenReserve) (Pool.sol#1120)
Reentrancy in Pool.mint(uint256,uint256) (Pool.sol#1258-1281):
  External calls:
  - WethUtils.KNIGHT.safeTransferFrom(_sender,address(this),_ethIn) (Pool.sol#1265)
  - WethUtils.KNIGHT.safeIncreaseAllowance(address(swapStrategy), wethSwapIn) (Pool.sol#1268)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
Pool.refreshCollateralRatio() (Pool.sol#1228-1254) uses timestamp for comparisons
  Dangerous comparisons:
  - require(bool,string)(block.timestamp - lastRefreshCrTimestamp >= refreshCooldown,Pool::refreshCollateralRatio: Must wait for the refresh cooldown since last refresh) (Pool.sol#1230)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (Pool.sol#443-463) uses assembly
- INLINE ASM (Pool.sol#455-458)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Pool.refreshCollateralRatio() (Pool.sol#1228-1254) compares to a boolean constant:
- require(bool,string)(collateralRatioPaused == false,Pool::refreshCollateralRatio: Collateral Ratio has been paused) (Pool.sol#1229)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality
INFO:Detectors:
Address.functionCall(address,bytes) (Pool.sol#327-329) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (Pool.sol#356-362) is never used and should be removed
INFO:Detectors:
Pool.priceTarget (Pool.sol#1095) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
name() should be declared external:
- ERC20.name() (Pool.sol#625-627)
symbol() should be declared external:
- ERC20.symbol() (Pool.sol#633-635)
decimals() should be declared external:
- ERC20.decimals() (Pool.sol#650-652)
totalSupply() should be declared external:
- ERC20.totalSupply() (Pool.sol#657-659)
balanceOf(address) should be declared external:
- ERC20.balanceOf(address) (Pool.sol#664-666)
transfer(address,uint256) should be declared external:
- ERC20.transfer(address,uint256) (Pool.sol#676-680)
approve(address,uint256) should be declared external:
- ERC20.approve(address,uint256) (Pool.sol#699-703)
transferFrom(address,address,uint256) should be declared external:
- ERC20.transferFrom(address,address,uint256) (Pool.sol#721-730)
increaseAllowance(address,uint256) should be declared external:
- ERC20.increaseAllowance(address,uint256) (Pool.sol#744-748)
decreaseAllowance(address,uint256) should be declared external:
- ERC20.decreaseAllowance(address,uint256) (Pool.sol#764-773)
renounceOwnership() should be declared external:
- Ownable.renounceOwnership() (Pool.sol#982-984)
transferOwnership(address) should be declared external:
- Ownable.transferOwnership(address) (Pool.sol#990-993)
refreshCollateralRatio() should be declared external:
- Pool.refreshCollateralRatio() (Pool.sol#1228-1254)
toggle(bool,bool) should be declared external:
- Pool.toggle(bool,bool) (Pool.sol#1396-1400)
setCollateralRatioOptions(uint256,uint256,uint256,uint256) should be declared external:
- Pool.setCollateralRatioOptions(uint256,uint256,uint256,uint256) (Pool.sol#1407-1418)
setCollateralRatioOptions(uint256,uint256,uint256,uint256) should be declared external:
- Pool.setCollateralRatioOptions(uint256,uint256,uint256,uint256) (Pool.sol#1407-1418)
toggleCollateralRatio(bool) should be declared external:
- Pool.toggleCollateralRatio(bool) (Pool.sol#1422-1427)
setFees(uint256,uint256) should be declared external:
- Pool.setFees(uint256,uint256) (Pool.sol#1432-1438)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:Pool.sol analyzed (18 contracts with 75 detectors), 92 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

## Slither log >> SwapStrategyPOL.sol

```
INFO:Detectors:
SwapStrategyPOL.addLiquidity(uint256,uint256,uint256) (SwapStrategyPOL.sol#763-786) uses timestamp for comparisons
  Dangerous comparisons:
  - yTokenAmt > 0 && wethAmt > 0 (SwapStrategyPOL.sol#769)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (SwapStrategyPOL.sol#482-502) uses assembly
- INLINE ASM (SwapStrategyPOL.sol#494-497)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Address.functionCall(address,bytes) (SwapStrategyPOL.sol#366-368) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (SwapStrategyPOL.sol#395-401) is never used and should be removed
Address.functionDelegateCall(address,bytes) (SwapStrategyPOL.sol#455-457) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (SwapStrategyPOL.sol#465-474) is never used and should be removed
Address.functionStaticCall(address,bytes) (SwapStrategyPOL.sol#428-430) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (SwapStrategyPOL.sol#438-447) is never used and should be removed
```

This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

Email: [audit@EtherAuthority.io](mailto:audit@EtherAuthority.io)

```

INFO:Detectors:
Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (SwapStrategyPOL.sol#89) is too similar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (SwapStrategyPOL.sol#90)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar
INFO:Detectors:
SwapStrategyPOL.slitherConstructorVariables() (SwapStrategyPOL.sol#685-803) uses literals with too many digits:
- swapSlippage = 200000 (SwapStrategyPOL.sol#693)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
renounceOwnership() should be declared external:
- Ownable.renounceOwnership() (SwapStrategyPOL.sol#660-662)
transferOwnership(address) should be declared external:
- Ownable.transferOwnership(address) (SwapStrategyPOL.sol#668-671)
lpBalance() should be declared external:
- SwapStrategyPOL.lpBalance() (SwapStrategyPOL.sol#712-714)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:SwapStrategyPOL.sol analyzed (13 contracts with 75 detectors), 42 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

## Slither log >> Timelock.sol

```

INFO:Detectors:
Timelock.constructor(address,uint256).admin_ (Timelock.sol#23) lacks a zero-check on :
- admin = admin_ (Timelock.sol#27)
Timelock.setPendingAdmin(address).pendingAdmin_ (Timelock.sol#47) lacks a zero-check on :
- pendingAdmin = pendingAdmin_ (Timelock.sol#49)
Timelock.executeTransaction(address,uint256,string,bytes,uint256).target (Timelock.sol#87) lacks a zero-check on :
- (success,returnData) = target.call{value: value}(callData) (Timelock.sol#111)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Reentrancy in Timelock.executeTransaction(address,uint256,string,bytes,uint256) (Timelock.sol#86-117):
  External calls:
  - (success,returnData) = target.call{value: value}(callData) (Timelock.sol#111)
  Event emitted after the call(s):
  - ExecuteTransaction(txHash,target,value,signature,data,eta) (Timelock.sol#114)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:

```

```

INFO:Detectors:
Low level call in Timelock.executeTransaction(address,uint256,string,bytes,uint256) (Timelock.sol#86-117):
- (success,returnData) = target.call{value: value}(callData) (Timelock.sol#111)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
setDelay(uint256) should be declared external:
- Timelock.setDelay(uint256) (Timelock.sol#31-38)
acceptAdmin() should be declared external:
- Timelock.acceptAdmin() (Timelock.sol#40-45)
setPendingAdmin(address) should be declared external:
- Timelock.setPendingAdmin(address) (Timelock.sol#47-52)
queueTransaction(address,uint256,string,bytes,uint256) should be declared external:
- Timelock.queueTransaction(address,uint256,string,bytes,uint256) (Timelock.sol#54-69)
cancelTransaction(address,uint256,string,bytes,uint256) should be declared external:
- Timelock.cancelTransaction(address,uint256,string,bytes,uint256) (Timelock.sol#71-84)
executeTransaction(address,uint256,string,bytes,uint256) should be declared external:
- Timelock.executeTransaction(address,uint256,string,bytes,uint256) (Timelock.sol#86-117)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:Timelock.sol analyzed (1 contracts with 75 detectors), 15 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

## Slither log >> HonorDaoChef.sol

```

INFO:Detectors:
renounceOwnership() should be declared external:
- Ownable.renounceOwnership() (HonorDaoChef.sol#597-599)
transferOwnership(address) should be declared external:
- Ownable.transferOwnership(address) (HonorDaoChef.sol#605-608)
poolLength() should be declared external:
- HonorDaoChef.poolLength() (HonorDaoChef.sol#673-675)
deposit(uint256,uint256,address) should be declared external:
- HonorDaoChef.deposit(uint256,uint256,address) (HonorDaoChef.sol#724-745)
withdraw(uint256,uint256,address) should be declared external:
- HonorDaoChef.withdraw(uint256,uint256,address) (HonorDaoChef.sol#751-772)
withdrawAndHarvest(uint256,uint256,address) should be declared external:
- HonorDaoChef.withdrawAndHarvest(uint256,uint256,address) (HonorDaoChef.sol#805-835)
emergencyWithdraw(uint256,address) should be declared external:
- HonorDaoChef.emergencyWithdraw(uint256,address) (HonorDaoChef.sol#840-854)
add(uint256,IERC20,IRewarder) should be declared external:
- HonorDaoChef.add(uint256,IERC20,IRewarder) (HonorDaoChef.sol#879-892)
set(uint256,uint256,IRewarder,bool) should be declared external:
- HonorDaoChef.set(uint256,uint256,IRewarder,bool) (HonorDaoChef.sol#899-912)
setRewardPerSecond(uint256) should be declared external:
- HonorDaoChef.setRewardPerSecond(uint256) (HonorDaoChef.sol#916-921)
getSlots(address,uint256) should be declared external:
- HonorDaoChef.getSlots(address,uint256) (HonorDaoChef.sol#941-952)
getTokenIds(address,uint256) should be declared external:
- HonorDaoChef.getTokenIds(address,uint256) (HonorDaoChef.sol#954-965)
depositNFT(address,uint256,uint256,uint256) should be declared external:
- HonorDaoChef.depositNFT(address,uint256,uint256,uint256) (HonorDaoChef.sol#967-988)
withdrawNFT(uint256,uint256) should be declared external:
- HonorDaoChef.withdrawNFT(uint256,uint256) (HonorDaoChef.sol#990-1002)
setNftController(address) should be declared external:
- HonorDaoChef.setNftController(address) (HonorDaoChef.sol#1004-1007)
withdrawNFT(uint256,uint256) should be declared external:
- HonorDaoChef.withdrawNFT(uint256,uint256) (HonorDaoChef.sol#990-1002)
setNftController(address) should be declared external:
- HonorDaoChef.setNftController(address) (HonorDaoChef.sol#1004-1007)
setNftBoostRate(uint256) should be declared external:
- HonorDaoChef.setNftBoostRate(uint256) (HonorDaoChef.sol#1009-1013)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:HonorDaoChef.sol analyzed (12 contracts with 75 detectors), 75 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

Email: [audit@EtherAuthority.io](mailto:audit@EtherAuthority.io)

## Slither log >> HonorDaoStaking.sol

```
INFO:Detectors:
HonorDaoStaking.userLocks (HonorDaoStaking.sol#834) is never used in HonorDaoStaking (HonorDaoStaking.sol#779-1268)
HonorDaoStaking.userEarnings (HonorDaoStaking.sol#835) is never used in HonorDaoStaking (HonorDaoStaking.sol#779-1268)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
INFO:Detectors:
HonorDaoStaking.lockedSupply (HonorDaoStaking.sol#830) should be constant
HonorDaoStaking.teamRewardPercent (HonorDaoStaking.sol#838) should be constant
HonorDaoStaking.totalSupply (HonorDaoStaking.sol#829) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
renounceOwnership() should be declared external:
- Ownable.renounceOwnership() (HonorDaoStaking.sol#711-713)
transferOwnership(address) should be declared external:
- Ownable.transferOwnership(address) (HonorDaoStaking.sol#719-722)
addReward(address,address) should be declared external:
- HonorDaoStaking.addReward(address,address) (HonorDaoStaking.sol#867-875)
withdrawableBalance(address) should be declared external:
- HonorDaoStaking.withdrawableBalance(address) (HonorDaoStaking.sol#1000-1018)
withdraw(uint256) should be declared external:
- HonorDaoStaking.withdraw(uint256) (HonorDaoStaking.sol#1071-1114)
getReward() should be declared external:
- HonorDaoStaking.getReward() (HonorDaoStaking.sol#1117-1133)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:HonorDaoStaking.sol analyzed (13 contracts with 75 detectors), 76 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

## Slither log >> HonorDaoZapMMSwap.sol

```
INFO:Detectors:
Pragma version0.8.4 (HonorDaoZapMMSwap.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (HonorDaoZapMMSwap.sol#344-349):
- (success) = recipient.call{value: amount}() (HonorDaoZapMMSwap.sol#347)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (HonorDaoZapMMSwap.sol#412-423):
- (success,returndata) = target.call{value: value}(data) (HonorDaoZapMMSwap.sol#421)
Low level call in Address.functionStaticCall(address,bytes,string) (HonorDaoZapMMSwap.sol#441-450):
- (success,returndata) = target.staticcall(data) (HonorDaoZapMMSwap.sol#448)
Low level call in Address.functionDelegateCall(address,bytes,string) (HonorDaoZapMMSwap.sol#468-477):
- (success,returndata) = target.delegatecall(data) (HonorDaoZapMMSwap.sol#475)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
INFO:Detectors:
Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (HonorDaoZapMMSwap.sol#92) is too similar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (HonorDaoZapMMSwap.sol#93)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar
INFO:Detectors:
renounceOwnership() should be declared external:
- Ownable.renounceOwnership() (HonorDaoZapMMSwap.sol#633-635)
transferOwnership(address) should be declared external:
- Ownable.transferOwnership(address) (HonorDaoZapMMSwap.sol#641-644)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:HonorDaoZapMMSwap.sol analyzed (13 contracts with 75 detectors), 45 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

## Slither log >> NFTController.sol

```
INFO:Detectors:
NFTController.initialize(address).owner (NFTController.sol#371) shadows:
- Ownable.owner() (NFTController.sol#342-344) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (NFTController.sol#201-221) uses assembly
- INLINE ASM (NFTController.sol#213-216)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
INFO:Detectors:
Pragma version0.8.4 (NFTController.sol#4) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (NFTController.sol#60-65):
- (success) = recipient.call{value: amount}() (NFTController.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (NFTController.sol#128-139):
- (success,returndata) = target.call{value: value}(data) (NFTController.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (NFTController.sol#157-166):
- (success,returndata) = target.staticcall(data) (NFTController.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (NFTController.sol#184-193):
- (success,returndata) = target.delegatecall(data) (NFTController.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
owner() should be declared external:
- Ownable.owner() (NFTController.sol#342-344)
renounceOwnership() should be declared external:
- Ownable.renounceOwnership() (NFTController.sol#351-354)
transferOwnership(address) should be declared external:
- Ownable.transferOwnership(address) (NFTController.sol#356-360)
initialize(address) should be declared external:
- NFTController.initialize(address) (NFTController.sol#371-373)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:NFTController.sol analyzed (6 contracts with 75 detectors), 24 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

Email: [audit@EtherAuthority.io](mailto:audit@EtherAuthority.io)

## Slither log >> NFTControllerProxy.sol

```
INFO:Detectors:
NFTControllerProxy.constructor(address,address,bytes).admin (NFTControllerProxy.sol#654) shadows:
- TransparentUpgradeableProxy.admin() (NFTControllerProxy.sol#587-589) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
Modifier TransparentUpgradeableProxy.ifAdmin() (NFTControllerProxy.sol#570-576) does not always execute _; or revertReference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-modifier

INFO:Detectors:
Pragma version0.8.4 (NFTControllerProxy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (NFTControllerProxy.sol#56-61):
- (success) = recipient.call{value: amount}() (NFTControllerProxy.sol#59)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (NFTControllerProxy.sol#124-135):
- (success,returndata) = target.call{value: value}(data) (NFTControllerProxy.sol#133)
Low level call in Address.functionStaticCall(address,bytes,string) (NFTControllerProxy.sol#153-162):
- (success,returndata) = target.staticcall(data) (NFTControllerProxy.sol#160)
Low level call in Address.functionDelegateCall(address,bytes,string) (NFTControllerProxy.sol#180-189):
- (success,returndata) = target.delegatecall(data) (NFTControllerProxy.sol#187)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Slither:NFTControllerProxy.sol analyzed (9 contracts with 75 detectors), 36 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

## Slither log >> DaoFund.sol

```
INFO:Detectors:
Fund.vestedBalance() (DaoFund.sol#565-576) uses timestamp for comparisons
Dangerous comparisons:
- block.timestamp <= _start (DaoFund.sol#569)
- block.timestamp > _start + _duration (DaoFund.sol#572)
Fund.transfer(address,uint256) (DaoFund.sol#583-590) uses timestamp for comparisons
Dangerous comparisons:
- require(bool,string)(amount <= claimable(),Fund::transfer: > vestedAmount) (DaoFund.sol#586)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (DaoFund.sol#273-293) uses assembly
- INLINE ASM (DaoFund.sol#285-288)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

INFO:Detectors:
Parameter Fund.initialize(address).yToken (DaoFund.sol#548) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
renounceOwnership() should be declared external:
- Ownable.renounceOwnership() (DaoFund.sol#422-424)
transferOwnership(address) should be declared external:
- Ownable.transferOwnership(address) (DaoFund.sol#430-433)
currentBalance() should be declared external:
- Fund.currentBalance() (DaoFund.sol#561-563)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:DaoFund.sol analyzed (8 contracts with 75 detectors), 27 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

## Slither log >> DevFund.sol

```
INFO:Detectors:
Fund.vestedBalance() (DevFund.sol#564-575) uses timestamp for comparisons
Dangerous comparisons:
- block.timestamp <= _start (DevFund.sol#568)
- block.timestamp > _start + _duration (DevFund.sol#571)
Fund.transfer(address,uint256) (DevFund.sol#582-589) uses timestamp for comparisons
Dangerous comparisons:
- require(bool,string)(amount <= claimable(),Fund::transfer: > vestedAmount) (DevFund.sol#585)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (DevFund.sol#272-292) uses assembly
- INLINE ASM (DevFund.sol#284-287)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

INFO:Detectors:
Parameter Fund.initialize(address).yToken (DevFund.sol#547) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
renounceOwnership() should be declared external:
- Ownable.renounceOwnership() (DevFund.sol#421-423)
transferOwnership(address) should be declared external:
- Ownable.transferOwnership(address) (DevFund.sol#429-432)
currentBalance() should be declared external:
- Fund.currentBalance() (DevFund.sol#560-562)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:DevFund.sol analyzed (8 contracts with 75 detectors), 27 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

## Slither log >> Fund.sol

```
INFO:Detectors:
Fund.vestedBalance() (Fund.sol#564-575) uses timestamp for comparisons
Dangerous comparisons:
- block.timestamp <= _start (Fund.sol#568)
- block.timestamp > _start + _duration (Fund.sol#571)
Fund.transfer(address,uint256) (Fund.sol#582-589) uses timestamp for comparisons
Dangerous comparisons:
- require(bool,string)(amount <= claimable(),Fund::transfer: > vestedAmount) (Fund.sol#585)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (Fund.sol#272-292) uses assembly
- INLINE ASM (Fund.sol#284-287)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

Email: [audit@EtherAuthority.io](mailto:audit@EtherAuthority.io)

```

INFO:Detectors:
Pragma version0.8.4 (Fund.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (Fund.sol#131-136):
- (success) = recipient.call{value: amount}() (Fund.sol#134)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (Fund.sol#199-210):
- (success,returndata) = target.call{value: value}(data) (Fund.sol#208)
Low level call in Address.functionStaticCall(address,bytes,string) (Fund.sol#228-237):
- (success,returndata) = target.staticcall(data) (Fund.sol#235)
Low level call in Address.functionDelegateCall(address,bytes,string) (Fund.sol#255-264):
- (success,returndata) = target.delegatecall(data) (Fund.sol#262)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Parameter Fund.initialize(address)._yToken (Fund.sol#547) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Fund (Fund.sol#540-591) does not implement functions:
- Fund.allocation() (Fund.sol#554)
- Fund.vestingDuration() (Fund.sol#558)
- Fund.vestingStart() (Fund.sol#556)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
INFO:Detectors:
renounceOwnership() should be declared external:
- Ownable.renounceOwnership() (Fund.sol#421-423)
transferOwnership(address) should be declared external:
- Ownable.transferOwnership(address) (Fund.sol#429-432)
currentBalance() should be declared external:
- Fund.currentBalance() (Fund.sol#560-562)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:Fund.sol analyzed (7 contracts with 75 detectors), 28 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

## Slither log >> HONORReserve.sol

```

INFO:Detectors:
HONORReserve.setRewarder(address)._rewarder (HONORReserve.sol#498) lacks a zero-check on :
- rewarder = _rewarder (HONORReserve.sol#500)
HONORReserve.setPool(address)._pool (HONORReserve.sol#504) lacks a zero-check on :
- pool = _pool (HONORReserve.sol#506)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (HONORReserve.sol#272-292) uses assembly
- INLINE ASM (HONORReserve.sol#284-287)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Pragma version0.8.4 (HONORReserve.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (HONORReserve.sol#131-136):
- (success) = recipient.call{value: amount}() (HONORReserve.sol#134)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (HONORReserve.sol#199-210):
- (success,returndata) = target.call{value: value}(data) (HONORReserve.sol#208)
Low level call in Address.functionStaticCall(address,bytes,string) (HONORReserve.sol#228-237):
- (success,returndata) = target.staticcall(data) (HONORReserve.sol#235)
Low level call in Address.functionDelegateCall(address,bytes,string) (HONORReserve.sol#255-264):
- (success,returndata) = target.delegatecall(data) (HONORReserve.sol#262)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Parameter HONORReserve.initialize(address)._honor (HONORReserve.sol#491) is not in mixedCase
Parameter HONORReserve.setRewarder(address)._rewarder (HONORReserve.sol#498) is not in mixedCase
Parameter HONORReserve.setPool(address)._pool (HONORReserve.sol#504) is not in mixedCase
Parameter HONORReserve.transfer(address,uint256)._to (HONORReserve.sol#510) is not in mixedCase
Parameter HONORReserve.transfer(address,uint256)._amount (HONORReserve.sol#510) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Slither:HONORReserve.sol analyzed (6 contracts with 75 detectors), 29 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

## Slither log >> TreasuryFund.sol

```

INFO:Detectors:
Fund.vestedBalance() (TreasuryFund.sol#564-575) uses timestamp for comparisons
Dangerous comparisons:
- block.timestamp <= _start (TreasuryFund.sol#568)
- block.timestamp > _start + _duration (TreasuryFund.sol#571)
Fund.transfer(address,uint256) (TreasuryFund.sol#582-589) uses timestamp for comparisons
Dangerous comparisons:
- require(bool,string)(amount <= claimable(),Fund::transfer: > vestedAmount) (TreasuryFund.sol#585)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (TreasuryFund.sol#272-292) uses assembly
- INLINE ASM (TreasuryFund.sol#284-287)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Parameter Fund.initialize(address)._yToken (TreasuryFund.sol#547) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
renounceOwnership() should be declared external:
- Ownable.renounceOwnership() (TreasuryFund.sol#421-423)
transferOwnership(address) should be declared external:
- Ownable.transferOwnership(address) (TreasuryFund.sol#429-432)
currentBalance() should be declared external:
- Fund.currentBalance() (TreasuryFund.sol#560-562)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:TreasuryFund.sol analyzed (8 contracts with 75 detectors), 27 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

Email: [audit@EtherAuthority.io](mailto:audit@EtherAuthority.io)

## Slither log >> MockERC20.sol

```
INFO:Detectors:
MockERC20.constructor(string,string,uint8)._name (MockERC20.sol#459) shadows:
- ERC20._name (MockERC20.sol#112) (state variable)
MockERC20.constructor(string,string,uint8)._symbol (MockERC20.sol#460) shadows:
- ERC20._symbol (MockERC20.sol#113) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
Context._msgData() (MockERC20.sol#100-102) is never used and should be removed
ERC20._burn(address,uint256) (MockERC20.sol#350-365) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.4 (MockERC20.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter MockERC20.mint(uint256)._amount (MockERC20.sol#466) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
name() should be declared external:
- ERC20.name() (MockERC20.sol#132-134)
symbol() should be declared external:
- ERC20.symbol() (MockERC20.sol#140-142)
decimals() should be declared external:
- ERC20.decimals() (MockERC20.sol#157-159)
- MockERC20.decimals() (MockERC20.sol#470-472)
totalSupply() should be declared external:
- ERC20.totalSupply() (MockERC20.sol#164-166)
balanceOf(address) should be declared external:
- ERC20.balanceOf(address) (MockERC20.sol#171-173)
transfer(address,uint256) should be declared external:
- ERC20.transfer(address,uint256) (MockERC20.sol#183-187)
approve(address,uint256) should be declared external:
- ERC20.approve(address,uint256) (MockERC20.sol#206-210)
approve(address,uint256) should be declared external:
- ERC20.approve(address,uint256) (MockERC20.sol#206-210)
transferFrom(address,address,uint256) should be declared external:
- ERC20.transferFrom(address,address,uint256) (MockERC20.sol#228-237)
increaseAllowance(address,uint256) should be declared external:
- ERC20.increaseAllowance(address,uint256) (MockERC20.sol#251-255)
decreaseAllowance(address,uint256) should be declared external:
- ERC20.decreaseAllowance(address,uint256) (MockERC20.sol#271-280)
mint(uint256) should be declared external:
- MockERC20.mint(uint256) (MockERC20.sol#466-468)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:MockERC20.sol analyzed (5 contracts with 75 detectors), 18 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

## Slither log >> MockTreasury.sol

```
INFO:Detectors:
Pragma version0.8.4 (MockTreasury.sol#1) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter MockTreasury.mock(uint256,uint256,uint256,uint256)._maxKnightxSupply (MockTreasury.sol#10) is not in mixedCase
Parameter MockTreasury.mock(uint256,uint256,uint256,uint256)._cr (MockTreasury.sol#11) is not in mixedCase
Parameter MockTreasury.mock(uint256,uint256,uint256,uint256)._mintingFee (MockTreasury.sol#12) is not in mixedCase
Parameter MockTreasury.mock(uint256,uint256,uint256,uint256)._redeemFee (MockTreasury.sol#13) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
mock(uint256,uint256,uint256,uint256) should be declared external:
- MockTreasury.mock(uint256,uint256,uint256,uint256) (MockTreasury.sol#9-19)
info() should be declared external:
- MockTreasury.info() (MockTreasury.sol#21-31)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:MockTreasury.sol analyzed (1 contracts with 75 detectors), 8 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

## Slither log >> MasterOracle.sol

```
INFO:Detectors:
Context._msgData() (MasterOracle.sol#18-20) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version0.8.4 (MasterOracle.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Variable MasterOracle.constructor(address,address,address,address)._oracleXToken (MasterOracle.sol#93) is too similar to MasterOracle.constructor(address,address,address,address)._oracleYToken (MasterOracle.sol#94)
Variable MasterOracle.oracleXToken (MasterOracle.sol#84) is too similar to MasterOracle.oracleYToken (MasterOracle.sol#85)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar
INFO:Detectors:
renounceOwnership() should be declared external:
- Ownable.renounceOwnership() (MasterOracle.sol#57-59)
transferOwnership(address) should be declared external:
- Ownable.transferOwnership(address) (MasterOracle.sol#65-68)
getXTokenPrice() should be declared external:
- MasterOracle.getXTokenPrice() (MasterOracle.sol#106-108)
getYTokenPrice() should be declared external:
- MasterOracle.getYTokenPrice() (MasterOracle.sol#110-112)
getXTokenTWAP() should be declared external:
- MasterOracle.getXTokenTWAP() (MasterOracle.sol#114-116)
getYTokenTWAP() should be declared external:
- MasterOracle.getYTokenTWAP() (MasterOracle.sol#118-120)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:MasterOracle.sol analyzed (4 contracts with 75 detectors), 11 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

Email: [audit@EtherAuthority.io](mailto:audit@EtherAuthority.io)

## Slither log >> UniswapPairOracle.sol

```
INFO:Detectors:
UniswapPairOracle.twap(address,uint256) (UniswapPairOracle.sol#730-745) uses timestamp for comparisons
  Dangerous comparisons:
    - require(bool,string)(timeElapsed <= PERIOD + LENIENCY,PairOracle::twap: Oracle was staled) (UniswapPairOracle.sol#732)
UniswapPairOracle.currentCumulativePrices(address) (UniswapPairOracle.sol#770-802) uses timestamp for comparisons
  Dangerous comparisons:
    - _blockTimestampLast != blockTimestamp (UniswapPairOracle.sol#788)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Babylonian.sqrt(uint256) (UniswapPairOracle.sol#19-31) is never used and should be removed
Context._msgData() (UniswapPairOracle.sol#250-252) is never used and should be removed
ERC20._burn(address,uint256) (UniswapPairOracle.sol#558-573) is never used and should be removed
ERC20._mint(address,uint256) (UniswapPairOracle.sol#535-545) is never used and should be removed
FixedPoint.decode(FixedPoint.uq112x112) (UniswapPairOracle.sol#83-85) is never used and should be removed
FixedPoint.div(FixedPoint.uq112x112,uint112) (UniswapPairOracle.sol#62-65) is never used and should be removed
FixedPoint.encode(uint112) (UniswapPairOracle.sol#52-54) is never used and should be removed
FixedPoint.encode144(uint144) (UniswapPairOracle.sol#57-59) is never used and should be removed
FixedPoint.reciprocal(FixedPoint.uq112x112) (UniswapPairOracle.sol#93-96) is never used and should be removed
FixedPoint.sqrt(FixedPoint.uq112x112) (UniswapPairOracle.sol#99-101) is never used and should be removed

INFO:Detectors:
renounceOwnership() should be declared external:
  - Ownable.renounceOwnership() (UniswapPairOracle.sol#289-291)
transferOwnership(address) should be declared external:
  - Ownable.transferOwnership(address) (UniswapPairOracle.sol#297-300)
name() should be declared external:
  - ERC20.name() (UniswapPairOracle.sol#340-342)
symbol() should be declared external:
  - ERC20.symbol() (UniswapPairOracle.sol#348-350)
decimals() should be declared external:
  - ERC20.decimals() (UniswapPairOracle.sol#365-367)
totalSupply() should be declared external:
  - ERC20.totalSupply() (UniswapPairOracle.sol#372-374)
balanceOf(address) should be declared external:
  - ERC20.balanceOf(address) (UniswapPairOracle.sol#379-381)
transfer(address,uint256) should be declared external:
  - ERC20.transfer(address,uint256) (UniswapPairOracle.sol#391-395)
approve(address,uint256) should be declared external:
  - ERC20.approve(address,uint256) (UniswapPairOracle.sol#414-418)
transferFrom(address,address,uint256) should be declared external:
  - ERC20.transferFrom(address,address,uint256) (UniswapPairOracle.sol#436-445)
increaseAllowance(address,uint256) should be declared external:
  - ERC20.increaseAllowance(address,uint256) (UniswapPairOracle.sol#459-463)
decreaseAllowance(address,uint256) should be declared external:
  - ERC20.decreaseAllowance(address,uint256) (UniswapPairOracle.sol#479-488)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:UniswapPairOracle.sol analyzed (10 contracts with 75 detectors), 48 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

## Slither log >> XToken.sol

```
INFO:Detectors:
Parameter XToken.setMinter(address)._minter (XToken.sol#503) is not in mixedCase
Parameter XToken._mint(address,uint256)._address (XToken.sol#511) is not in mixedCase
Parameter XToken._mint(address,uint256)._amount (XToken.sol#511) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
name() should be declared external:
  - ERC20.name() (XToken.sol#133-135)
symbol() should be declared external:
  - ERC20.symbol() (XToken.sol#141-143)
decimals() should be declared external:
  - ERC20.decimals() (XToken.sol#158-160)
totalSupply() should be declared external:
  - ERC20.totalSupply() (XToken.sol#165-167)
balanceOf(address) should be declared external:
  - ERC20.balanceOf(address) (XToken.sol#172-174)
transfer(address,uint256) should be declared external:
  - ERC20.transfer(address,uint256) (XToken.sol#184-188)
approve(address,uint256) should be declared external:
  - ERC20.approve(address,uint256) (XToken.sol#207-211)
transferFrom(address,address,uint256) should be declared external:
  - ERC20.transferFrom(address,address,uint256) (XToken.sol#229-238)
increaseAllowance(address,uint256) should be declared external:
  - ERC20.increaseAllowance(address,uint256) (XToken.sol#252-256)
decreaseAllowance(address,uint256) should be declared external:
  - ERC20.decreaseAllowance(address,uint256) (XToken.sol#272-281)
burn(uint256) should be declared external:
  - ERC20Burnable.burn(uint256) (XToken.sol#462-464)
burnFrom(address,uint256) should be declared external:
  - ERC20Burnable.burnFrom(address,uint256) (XToken.sol#477-480)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:XToken.sol analyzed (6 contracts with 75 detectors), 21 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

## Slither log >> YToken.sol

```
INFO:Detectors:
YToken.constructor(string,string)._name (YToken.sol#485) shadows:
  - ERC20._name (YToken.sol#113) (state variable)
YToken.constructor(string,string)._symbol (YToken.sol#485) shadows:
  - ERC20._symbol (YToken.sol#114) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
Context._msgData() (YToken.sol#101-103) is never used and should be removed
ERC20._mint(address,uint256) (YToken.sol#328-338) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version0.8.4 (YToken.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
```

This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

Email: [audit@EtherAuthority.io](mailto:audit@EtherAuthority.io)

```

INFO:Detectors:
name() should be declared external:
- ERC20.name() (YToken.sol#133-135)
symbol() should be declared external:
- ERC20.symbol() (YToken.sol#141-143)
decimals() should be declared external:
- ERC20.decimals() (YToken.sol#158-160)
totalSupply() should be declared external:
- ERC20.totalSupply() (YToken.sol#165-167)
balanceOf(address) should be declared external:
- ERC20.balanceOf(address) (YToken.sol#172-174)
transfer(address,uint256) should be declared external:
- ERC20.transfer(address,uint256) (YToken.sol#184-188)
approve(address,uint256) should be declared external:
- ERC20.approve(address,uint256) (YToken.sol#207-211)
transferFrom(address,address,uint256) should be declared external:
- ERC20.transferFrom(address,address,uint256) (YToken.sol#229-238)
increaseAllowance(address,uint256) should be declared external:
- ERC20.increaseAllowance(address,uint256) (YToken.sol#252-256)
decreaseAllowance(address,uint256) should be declared external:
- ERC20.decreaseAllowance(address,uint256) (YToken.sol#272-281)
burn(uint256) should be declared external:
- ERC20Burnable.burn(uint256) (YToken.sol#462-464)
burnFrom(address,uint256) should be declared external:
- ERC20Burnable.burnFrom(address,uint256) (YToken.sol#477-480)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:YToken.sol analyzed (6 contracts with 75 detectors), 18 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

## Slither log >> HONOR.sol

```

INFO:Detectors:
YToken.constructor(string,string)._name (HONOR.sol#485) shadows:
- ERC20._name (HONOR.sol#113) (state variable)
YToken.constructor(string,string)._symbol (HONOR.sol#485) shadows:
- ERC20._symbol (HONOR.sol#114) (state variable)
HONOR.constructor(string,string,address,address,address,address)._name (HONOR.sol#497) shadows:
- ERC20._name (HONOR.sol#113) (state variable)
HONOR.constructor(string,string,address,address,address,address)._symbol (HONOR.sol#498) shadows:
- ERC20._symbol (HONOR.sol#114) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
Context._msgData() (HONOR.sol#101-103) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version0.8.4 (HONOR.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Function HONOR.OpenTrade() (HONOR.sol#512-516) is not in mixedCase
Parameter HONOR.includeToWhitelist(address)._address (HONOR.sol#518) is not in mixedCase
Parameter HONOR.excludeFromWhitelist(address)._address (HONOR.sol#525) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

```

```

INFO:Detectors:
name() should be declared external:
- ERC20.name() (HONOR.sol#133-135)
symbol() should be declared external:
- ERC20.symbol() (HONOR.sol#141-143)
decimals() should be declared external:
- ERC20.decimals() (HONOR.sol#158-160)
totalSupply() should be declared external:
- ERC20.totalSupply() (HONOR.sol#165-167)
balanceOf(address) should be declared external:
- ERC20.balanceOf(address) (HONOR.sol#172-174)
transfer(address,uint256) should be declared external:
- ERC20.transfer(address,uint256) (HONOR.sol#184-188)
approve(address,uint256) should be declared external:
- ERC20.approve(address,uint256) (HONOR.sol#207-211)
transferFrom(address,address,uint256) should be declared external:
- ERC20.transferFrom(address,address,uint256) (HONOR.sol#229-238)
increaseAllowance(address,uint256) should be declared external:
- ERC20.increaseAllowance(address,uint256) (HONOR.sol#252-256)
decreaseAllowance(address,uint256) should be declared external:
- ERC20.decreaseAllowance(address,uint256) (HONOR.sol#272-281)
burn(uint256) should be declared external:
- ERC20Burnable.burn(uint256) (HONOR.sol#462-464)
burnFrom(address,uint256) should be declared external:
- ERC20Burnable.burnFrom(address,uint256) (HONOR.sol#477-480)
includeToWhitelist(address) should be declared external:
- HONOR.includeToWhitelist(address) (HONOR.sol#518-523)
excludeFromWhitelist(address) should be declared external:
- HONOR.excludeFromWhitelist(address) (HONOR.sol#525-530)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:HONOR.sol analyzed (7 contracts with 75 detectors), 24 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

## Slither log >> KNIGHTTestToken.sol

```

INFO:Detectors:
XToken.constructor(string,string)._name (KNIGHTTestToken.sol#497) shadows:
- ERC20._name (KNIGHTTestToken.sol#113) (state variable)
XToken.constructor(string,string)._symbol (KNIGHTTestToken.sol#497) shadows:
- ERC20._symbol (KNIGHTTestToken.sol#114) (state variable)
KnightTestToken.constructor(string,string)._name (KNIGHTTestToken.sol#520) shadows:
- ERC20._name (KNIGHTTestToken.sol#113) (state variable)
KnightTestToken.constructor(string,string)._symbol (KNIGHTTestToken.sol#520) shadows:
- ERC20._symbol (KNIGHTTestToken.sol#114) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
XToken.setMinter(address)._minter (KNIGHTTestToken.sol#503) lacks a zero-check on :
- minter = _minter (KNIGHTTestToken.sol#505)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

```

This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

Email: [audit@EtherAuthority.io](mailto:audit@EtherAuthority.io)

```

INFO:Detectors:
Context._msgData() (KNIGHTTestToken.sol#101-103) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version0.8.4 (KNIGHTTestToken.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter XToken.setMinter(address)._minter (KNIGHTTestToken.sol#503) is not in mixedCase
Parameter XToken.mint(address,uint256)._address (KNIGHTTestToken.sol#511) is not in mixedCase
Parameter XToken.mint(address,uint256)._amount (KNIGHTTestToken.sol#511) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
name() should be declared external:
- ERC20.name() (KNIGHTTestToken.sol#133-135)
symbol() should be declared external:
- ERC20.symbol() (KNIGHTTestToken.sol#141-143)
decimals() should be declared external:
- ERC20.decimals() (KNIGHTTestToken.sol#158-160)
totalSupply() should be declared external:
- ERC20.totalSupply() (KNIGHTTestToken.sol#165-167)
balanceOf(address) should be declared external:
- ERC20.balanceOf(address) (KNIGHTTestToken.sol#172-174)
transfer(address,uint256) should be declared external:
- ERC20.transfer(address,uint256) (KNIGHTTestToken.sol#184-188)
approve(address,uint256) should be declared external:
- ERC20.approve(address,uint256) (KNIGHTTestToken.sol#207-211)
transferFrom(address,address,uint256) should be declared external:
- ERC20.transferFrom(address,address,uint256) (KNIGHTTestToken.sol#229-238)
increaseAllowance(address,uint256) should be declared external:
- ERC20.increaseAllowance(address,uint256) (KNIGHTTestToken.sol#252-256)
decreaseAllowance(address,uint256) should be declared external:
- ERC20.decreaseAllowance(address,uint256) (KNIGHTTestToken.sol#272-281)
burn(uint256) should be declared external:
- ERC20Burnable.burn(uint256) (KNIGHTTestToken.sol#462-464)
decreaseAllowance(address,uint256) should be declared external:
- ERC20.decreaseAllowance(address,uint256) (KNIGHTTestToken.sol#272-281)
burn(uint256) should be declared external:
- ERC20Burnable.burn(uint256) (KNIGHTTestToken.sol#462-464)
burnFrom(address,uint256) should be declared external:
- ERC20Burnable.burnFrom(address,uint256) (KNIGHTTestToken.sol#477-480)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:KNIGHTTestToken.sol analyzed (7 contracts with 75 detectors), 23 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

## Slither log >> XNIGHT.sol

```

INFO:Detectors:
XToken.constructor(string,string)._name (XNIGHT.sol#497) shadows:
- ERC20._name (XNIGHT.sol#113) (state variable)
XToken.constructor(string,string)._symbol (XNIGHT.sol#497) shadows:
- ERC20._symbol (XNIGHT.sol#114) (state variable)
XNIGHT.constructor(string,string)._name (XNIGHT.sol#524) shadows:
- ERC20._name (XNIGHT.sol#113) (state variable)
XNIGHT.constructor(string,string)._symbol (XNIGHT.sol#524) shadows:
- ERC20._symbol (XNIGHT.sol#114) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
XToken.setMinter(address)._minter (XNIGHT.sol#503) lacks a zero-check on :
- minter = _minter (XNIGHT.sol#505)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Context._msgData() (XNIGHT.sol#101-103) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version0.8.4 (XNIGHT.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
name() should be declared external:
- ERC20.name() (XNIGHT.sol#133-135)
symbol() should be declared external:
- ERC20.symbol() (XNIGHT.sol#141-143)
decimals() should be declared external:
- ERC20.decimals() (XNIGHT.sol#158-160)
totalSupply() should be declared external:
- ERC20.totalSupply() (XNIGHT.sol#165-167)
balanceOf(address) should be declared external:
- ERC20.balanceOf(address) (XNIGHT.sol#172-174)
transfer(address,uint256) should be declared external:
- ERC20.transfer(address,uint256) (XNIGHT.sol#184-188)
approve(address,uint256) should be declared external:
- ERC20.approve(address,uint256) (XNIGHT.sol#207-211)
transferFrom(address,address,uint256) should be declared external:
- ERC20.transferFrom(address,address,uint256) (XNIGHT.sol#229-238)
increaseAllowance(address,uint256) should be declared external:
- ERC20.increaseAllowance(address,uint256) (XNIGHT.sol#252-256)
decreaseAllowance(address,uint256) should be declared external:
- ERC20.decreaseAllowance(address,uint256) (XNIGHT.sol#272-281)
burn(uint256) should be declared external:
- ERC20Burnable.burn(uint256) (XNIGHT.sol#462-464)
burnFrom(address,uint256) should be declared external:
- ERC20Burnable.burnFrom(address,uint256) (XNIGHT.sol#477-480)
includeToWhitelist(address) should be declared external:
- XNIGHT.includeToWhitelist(address) (XNIGHT.sol#535-540)
excludeFromWhitelist(address) should be declared external:
- XNIGHT.excludeFromWhitelist(address) (XNIGHT.sol#542-547)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:XNIGHT.sol analyzed (7 contracts with 75 detectors), 28 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

Email: [audit@EtherAuthority.io](mailto:audit@EtherAuthority.io)

## Slither log >> HonorDaoTreasury.sol

```
INFO:Detectors:
Pragma version0.8.4 (HonorDaoTreasury.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (HonorDaoTreasury.sol#136-141):
- (success) = recipient.call{value: amount}{} (HonorDaoTreasury.sol#139)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (HonorDaoTreasury.sol#204-215):
- (success, returndata) = target.call{value: value}(data) (HonorDaoTreasury.sol#213)
Low level call in Address.functionStaticCall(address,bytes,string) (HonorDaoTreasury.sol#233-242):
- (success, returndata) = target.staticcall(data) (HonorDaoTreasury.sol#240)
Low level call in Address.functionDelegateCall(address,bytes,string) (HonorDaoTreasury.sol#260-269):
- (success, returndata) = target.delegatecall(data) (HonorDaoTreasury.sol#267)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
HonorDaoTreasury (HonorDaoTreasury.sol#462-537) should inherit from IHonorDaoTreasury (HonorDaoTreasury.sol#4-8)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-inheritance
INFO:Detectors:
Parameter HonorDaoTreasury.balanceOf(address)._token (HonorDaoTreasury.sol#479) is not in mixedCase
Parameter HonorDaoTreasury.requestFund(address,uint256)._token (HonorDaoTreasury.sol#488) is not in mixedCase
Parameter HonorDaoTreasury.requestFund(address,uint256)._amount (HonorDaoTreasury.sol#488) is not in mixedCase
Parameter HonorDaoTreasury.addStrategy(address)._strategy (HonorDaoTreasury.sol#497) is not in mixedCase
Parameter HonorDaoTreasury.removeStrategy(address)._strategy (HonorDaoTreasury.sol#507) is not in mixedCase
Parameter HonorDaoTreasury.allocateFee(address,uint256)._token (HonorDaoTreasury.sol#524) is not in mixedCase
Parameter HonorDaoTreasury.allocateFee(address,uint256)._amount (HonorDaoTreasury.sol#524) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
renounceOwnership() should be declared external:
- Ownable.renounceOwnership() (HonorDaoTreasury.sol#426-428)
transferOwnership(address) should be declared external:
- Ownable.transferOwnership(address) (HonorDaoTreasury.sol#434-437)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:HonorDaoTreasury.sol analyzed (8 contracts with 75 detectors), 31 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

## Slither log >> StratRecollateralize.sol

```
INFO:Detectors:
Pragma version0.8.4 (StratRecollateralize.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (StratRecollateralize.sol#146-151):
- (success) = recipient.call{value: amount}{} (StratRecollateralize.sol#149)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (StratRecollateralize.sol#214-225):
- (success, returndata) = target.call{value: value}(data) (StratRecollateralize.sol#223)
Low level call in Address.functionStaticCall(address,bytes,string) (StratRecollateralize.sol#243-252):
- (success, returndata) = target.staticcall(data) (StratRecollateralize.sol#250)
Low level call in Address.functionDelegateCall(address,bytes,string) (StratRecollateralize.sol#270-279):
- (success, returndata) = target.delegatecall(data) (StratRecollateralize.sol#277)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Constant WethUtils.weth (StratRecollateralize.sol#396) is not in UPPER_CASE_WITH_UNDERSCORES
Parameter StratRecollateralize.recollateralize(uint256)._amount (StratRecollateralize.sol#498) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
renounceOwnership() should be declared external:
- Ownable.renounceOwnership() (StratRecollateralize.sol#460-462)
transferOwnership(address) should be declared external:
- Ownable.transferOwnership(address) (StratRecollateralize.sol#468-471)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:StratRecollateralize.sol analyzed (10 contracts with 75 detectors), 27 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

## Slither log >> StratReduceReserveLP.sol

```
INFO:Detectors:
StratReduceReserveLP.constructor(IERC20,address,IERC20,IUniswapV2Router02,address[],IHonorDaoTreasury)._yTokenFund (StratReduceReserveLP.sol#630) lacks a zero-check on :
- yTokenFund = _yTokenFund (StratReduceReserveLP.sol#638)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Reentrancy in StratReduceReserveLP.reduceReserve(uint256,uint256) (StratReduceReserveLP.sol#648-667):
  External calls:
  - treasury.requestFund(address(lp),_amount) (StratReduceReserveLP.sol#652)
  - lp.safeTransferFrom(address(treasury),address(this),_amount) (StratReduceReserveLP.sol#653)
  - lp.safeIncreaseAllowance(address(swapRouter),_amount) (StratReduceReserveLP.sol#656)
  - swapRouter.removeLiquidity(address(yToken),address(WethUtils.KNIGHT),_amount,0,0,address(this),block.timestamp) (StratReduceReserveLP.sol#657)
```

```
INFO:Detectors:
Function IUniswapV2Router01.WETH() (StratReduceReserveLP.sol#14) is not in mixedCase
Constant WethUtils.weth (StratReduceReserveLP.sol#529) is not in UPPER_CASE_WITH_UNDERSCORES
Parameter StratReduceReserveLP.reduceReserve(uint256,uint256)._amount (StratReduceReserveLP.sol#648) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (StratReduceReserveLP.sol#19) is too similar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (StratReduceReserveLP.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar
INFO:Detectors:
renounceOwnership() should be declared external:
- Ownable.renounceOwnership() (StratReduceReserveLP.sol#593-595)
transferOwnership(address) should be declared external:
- Ownable.transferOwnership(address) (StratReduceReserveLP.sol#601-604)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:StratReduceReserveLP.sol analyzed (11 contracts with 75 detectors), 33 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

Email: [audit@EtherAuthority.io](mailto:audit@EtherAuthority.io)

# Solidity Static Analysis

## Pool.sol

### Security

#### Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in Pool.redeem(uint256,uint256,uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 1283:4:

#### 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.

[more](#)

Pos: 1252:33:

### Gas & Economy

#### Gas costs:

Gas requirement of function Pool.collect is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 1323:4:

### Miscellaneous

#### Constant/View/Pure functions:

Pool.transferToTreasury(uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 1490:4:

#### Similar variable names:

Pool(address,address,address) : Variables have very similar names "xToken" and "\_yToken". Note: Modifiers are currently not considered by this static analysis.

Pos: 1119:25:

#### 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.

[more](#)

Pos: 1230:8:

### Data truncated:

Division of integer values yields an integer value again. That means e.g.  $10 / 100 = 0$  instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 1202:29:

## SwapStrategyPOL.sol

### Security

#### Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in `SwapStrategyPOL.addLiquidity(uint256,uint256,uint256)`: Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 763:4:

### Gas & Economy

#### Gas costs:

Gas requirement of function `SwapStrategyPOL.yToken` is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 688:4:

### ERC

#### ERC20:

ERC20 contract's "decimals" function should have "uint8" as return type

[more](#)

Pos: 37:4:

### Miscellaneous

#### Constant/View/Pure functions:

`WethUtils.transfer(address,uint256)` : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 612:4:

#### Similar variable names:

`SwapStrategyPOL.addLiquidity(uint256,uint256,uint256)` : Variables have very similar names "\_amountA" and "\_amountB". Note: Modifiers are currently not considered by this static analysis.

Pos: 784:54:

### 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.

[more](#)

Pos: 794:8:

### Data truncated:

Division of integer values yields an integer value again. That means e.g.  $10 / 100 = 0$  instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 771:34:

## Timelock.sol

### Security

#### Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in

Timelock.executeTransaction(address,uint256,string,bytes,uint256): Could potentially lead to re-entrancy vulnerability.

[more](#)

Pos: 86:4:

#### 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.

[more](#)

Pos: 120:15:

### Gas & Economy

#### Gas costs:

Gas requirement of function Timelock.queueTransaction is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 54:4:

### Miscellaneous

#### Similar variable names:

Timelock.(address,uint256) : Variables have very similar names "MINIMUM\_DELAY" and "MAXIMUM\_DELAY".

Pos: 24:26:

### 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.

[more](#)

Pos: 112:8:

## HonorDaoChef.sol

### Security

#### Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in HonorDaoChef.withdrawNFT(uint256,uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 990:4:

#### 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.

[more](#)

Pos: 890:72:

### Gas & Economy

#### Gas costs:

Gas requirement of function HonorDaoChef.pendingReward is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 681:4:

### Miscellaneous

#### 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.

[more](#)

Pos: 1010:8:

### Data truncated:

Division of integer values yields an integer value again. That means e.g.  $10 / 100 = 0$  instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 789:28:

## HonorDaoStaking.sol

### Security

#### Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in `HonorDaoStaking.getReward()`: Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 1117:4:

#### 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.

[more](#)

Pos: 1055:29:

### Gas & Economy

#### Gas costs:

Gas requirement of function `HonorDaoStaking.lockDuration` is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 816:4:

#### Delete dynamic array:

The "delete" operation when applied to a dynamically sized array in Solidity generates code to delete each of the elements contained. If the array is large, this operation can surpass the block gas limit and raise an OOG exception. Also nested dynamically sized objects can produce the same results.

[more](#)

Pos: 1138:8:

### For loop over 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.

[more](#)

Pos: 1118:8:

## Miscellaneous

### Constant/View/Pure functions:

HonorDaoStaking.lockedBalances(address) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 972:4:

### Similar variable names:

HonorDaoStaking.claimableRewards(address) : Variables have very similar names "balance" and "balances". Note: Modifiers are currently not considered by this static analysis.

Pos: 930:69:

### 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.

[more](#)

Pos: 1072:8:

## HonorDaoZapMMSwap.sol

### Security

### Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in HonorDaoZapMMSwap.swap(address,uint256,address): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 793:4:

## Gas & Economy

### Gas costs:

Gas requirement of function `HonorDaoZapMMSwap.swap` is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 738:4:

## ERC

### ERC20:

ERC20 contract's "decimals" function should have "uint8" as return type

[more](#)

Pos: 39:4:

## Miscellaneous

### Constant/View/Pure functions:

`HonorDaoZapMMSwap.approveToken(address,address,uint256)` : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 829:4:

### Similar variable names:

`HonorDaoZapMMSwap.swap(address,uint256,address)` : Variables have very similar names "\_token" and "\_token0". Note: Modifiers are currently not considered by this static analysis.

Pos: 798:8:

### 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.

[more](#)

Pos: 857:8:

### Data truncated:

Division of integer values yields an integer value again. That means e.g.  $10 / 100 = 0$  instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 848:15:

## NFTController.sol

### Security

#### Low level calls:

Use of "delegatecall": should be avoided whenever possible. External code, that is called can change the state of the calling contract and send ether from the caller's balance. If this is wanted behaviour, use the Solidity library feature if possible.

[more](#)

Pos: 191:50:

### Gas & Economy

#### Gas costs:

Gas requirement of function NFTController.getBoostRate is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 375:4:

### Miscellaneous

#### Constant/View/Pure functions:

Address.verifyCallResult(bool,bytes,string) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 201:4:

#### Similar variable names:

NFTController.setBoostRate(address,uint256,uint256) : Variables have very similar names "token" and "tokenId". Note: Modifiers are currently not considered by this static analysis.

Pos: 401:18:

#### 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.

[more](#)

Pos: 357:8:

## NFTControllerProxy.sol

### Security

### Low level calls:

Use of "delegatecall": should be avoided whenever possible. External code, that is called can change the state of the calling contract and send ether from the caller's balance. If this is wanted behaviour, use the Solidity library feature if possible.

[more](#)

Pos: 187:50:

## Gas & Economy

### Gas costs:

Gas requirement of function TransparentUpgradeableProxy.upgradeToAndCall is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 631:4:

## Miscellaneous

### 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.

[more](#)

Pos: 646:8:

## DaoFund.sol

### Security

### 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.

[more](#)

Pos: 575:31:

## Gas & Economy

### Gas costs:

Gas requirement of function DaoFund.transfer is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 583:4:

## Miscellaneous

### Constant/View/Pure functions:

SafeERC20.\_callOptionalReturn(contract IERC20,bytes) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 365:4:

### 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.

[more](#)

Pos: 586:8:

### Data truncated:

Division of integer values yields an integer value again. That means e.g.  $10 / 100 = 0$  instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 575:15:

## DevFund.sol

### Security

#### Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in SafeERC20.safeDecreaseAllowance(contract IERC20,address,uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 345:4:

### Gas & Economy

#### Gas costs:

Gas requirement of function DevFund.transfer is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 582:4:

### Miscellaneous

### 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.

[more](#)

Pos: 585:8:

### Data truncated:

Division of integer values yields an integer value again. That means e.g.  $10 / 100 = 0$  instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 574:15:

## Fund.sol

### Security

#### 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.

[more](#)

Pos: 574:31:

### Miscellaneous

#### Constant/View/Pure functions:

SafeERC20.\_callOptionalReturn(contract IERC20,bytes) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 364:4:

### 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.

[more](#)

Pos: 585:8:

### Data truncated:

Division of integer values yields an integer value again. That means e.g.  $10 / 100 = 0$  instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 574:15:

## HONORReserve.sol

### Security

#### Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in `SafeERC20.safeDecreaseAllowance(contract IERC20,address,uint256)`: Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 345:4:

### Gas & Economy

#### Gas costs:

Gas requirement of function `HONORReserve.transfer` is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 510:4:

### Miscellaneous

#### 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.

[more](#)

Pos: 512:8:

## MockTreasury.sol

### Security

#### 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.

[more](#)

Pos: 574:31:

### Gas & Economy

#### Gas costs:

Gas requirement of function `TreasuryFund.transfer` is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 582:4:

### Miscellaneous

### 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.

[more](#)

Pos: 585:8:

### Data truncated:

Division of integer values yields an integer value again. That means e.g.  $10 / 100 = 0$  instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 574:15:

## MockERC20.sol

### Security

#### Gas costs:

Gas requirement of function MockERC20.mint is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 466:4:

### Miscellaneous

#### Constant/View/Pure functions:

ERC20.\_afterTokenTransfer(address,address,uint256) : Potentially should be constant/view/pure but is not.

[more](#)

Pos: 448:4:

#### 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.

[more](#)

Pos: 407:12:

## TreasuryFund.sol

## Security

### 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.

[more](#)

Pos: 574:31:

## Gas & Economy

### Gas costs:

Gas requirement of function TreasuryFund.currentBalance is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 560:4:

## Miscellaneous

### Constant/View/Pure functions:

SafeERC20.\_callOptionalReturn(contract IERC20,bytes) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 364:4:

### 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.

[more](#)

Pos: 585:8:

### Data truncated:

Division of integer values yields an integer value again. That means e.g.  $10 / 100 = 0$  instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 574:15:

## MasterOracle.sol

## Gas & Economy

### Gas costs:

Gas requirement of function `MasterOracle.getYTokenTWAP` is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 118:4:

## Miscellaneous

### Similar variable names:

`MasterOracle.getYTokenTWAP()` : Variables have very similar names "xToken" and "yToken". Note: Modifiers are currently not considered by this static analysis.

Pos: 119:33:

### 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.

[more](#)

Pos: 99:8:

## UniswapPairOracle.sol

### Security

### 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.

[more](#)

Pos: 766:22:

## Gas & Economy

### Gas costs:

Gas requirement of function `UniswapPairOracle.spot` is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 747:4:

## ERC

### ERC20:

ERC20 contract's "decimals" function should have "uint8" as return type

[more](#)

Pos: 110:4:

## Miscellaneous

### Constant/View/Pure functions:

`UniswapPairOracle.currentCumulativePrices(address)` : Is constant but potentially should not be.

Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 770:4:

### Similar variable names:

`UniswapPairOracle.update()` : Variables have very similar names "price0CumulativeLast" and "price1CumulativeLast". Note: Modifiers are currently not considered by this static analysis.

Pos: 724:12:

### 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.

[more](#)

Pos: 753:8:

### Data truncated:

Division of integer values yields an integer value again. That means e.g.  $10 / 100 = 0$  instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 758:21:

## XToken.sol

### Gas & Economy

#### Gas costs:

Gas requirement of function ERC20.decreaseAllowance is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 272:4:

### Miscellaneous

#### Constant/View/Pure functions:

ERC20.\_afterTokenTransfer(address,address,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 449:4:

#### Similar variable names:

ERC20Burnable.burnFrom(address,uint256) : Variables have very similar names "account" and "amount". Note: Modifiers are currently not considered by this static analysis.

Pos: 479:23:

#### 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.

[more](#)

Pos: 504:8:

## YToken.sol

### Gas & Economy

#### Gas costs:

Gas requirement of function ERC20.decreaseAllowance is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 272:4:

### Miscellaneous

### Constant/View/Pure functions:

ERC20.\_afterTokenTransfer(address,address,uint256) : Potentially should be constant/view/pure but is not.

[more](#)

Pos: 449:4:

### Similar variable names:

ERC20Burnable.burnFrom(address,uint256) : Variables have very similar names "account" and "amount".

Pos: 479:23:

### 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.

[more](#)

Pos: 408:12:

## HONOR.sol

### Gas & Economy

#### Gas costs:

Gas requirement of function HONOR.burnFrom is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 477:4:

### Miscellaneous

#### Constant/View/Pure functions:

HONOR.\_transfer(address,address,uint256) : Potentially should be constant/view/pure but is not.

[more](#)

Pos: 532:4:

#### Similar variable names:

HONOR.(string,string,address,address,address,address) : Variables have very similar names "\_daoFund" and "\_devFund".

Pos: 506:14:

#### 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.

[more](#)

Pos: 537:8:

## KNIGHTTestToken.sol

### Gas & Economy

#### Gas costs:

Gas requirement of function KnightTestToken.mint is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)  
Pos: 511:4:

### Miscellaneous

#### 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.

[more](#)

Pos: 504:8:

## XNIGHT.sol

### Gas & Economy

#### Gas costs:

Gas requirement of function XNIGHT.mint is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)  
Pos: 511:4:

### Miscellaneous

#### 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.

[more](#)

Pos: 550:8:

## HonorDaoTreasury.sol

### Security

#### Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in  
HonorDaoTreasury.allocateFee(address,uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 524:4:

## Gas & Economy

### Gas costs:

Gas requirement of function `HonorDaoTreasury.balanceOf` is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Pos: 479:4:

### For loop over 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.

[more](#)

Pos: 511:8:

## Miscellaneous

### 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.

[more](#)

Pos: 526:8:

### Delete from 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.

[more](#)

Pos: 509:8:

## StratRecollateralize.sol

### Security

#### Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in `StratRecollateralize.recollateralize(uint256)`: Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 498:4:

## Gas & Economy

### Gas costs:

Gas requirement of function `StratRecollateralize.pool` is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)  
Pos: 488:4:

## Miscellaneous

### 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.

[more](#)

Pos: 499:8:

## StratReduceReserveLP.sol

## Security

### Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in `StratReduceReserveLP.reduceReserve(uint256,uint256)`: Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 648:4:

### 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.

[more](#)

Pos: 674:127:

## Gas & Economy

### Gas costs:

Gas requirement of function `StratReduceReserveLP.reduceReserve` is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)  
Pos: 648:4:

## Miscellaneous

### Constant/View/Pure functions:

WethUtils.transfer(address,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 544:4:

### 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.

[more](#)

Pos: 649:8:

### 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.

[more](#)

Pos: 650:8:

# Solhint Linter

## Pool.sol

```
Pool.sol:521:18: Error: Parse error: missing ';' at '{'  
Pool.sol:768:18: Error: Parse error: missing ';' at '{'  
Pool.sol:801:18: Error: Parse error: missing ';' at '{'  
Pool.sol:850:18: Error: Parse error: missing ';' at '{'  
Pool.sol:901:22: Error: Parse error: missing ';' at '{'
```

## SwapStrategyPOL.sol

```
SwapStrategyPOL.sol:560:18: Error: Parse error: missing ';' at '{'
```

## Timelock.sol

```
Timelock.sol:3:1: Error: Compiler version 0.8.4 does not satisfy the  
r semver requirement  
Timelock.sol:23:5: Error: Explicitly mark visibility in function (Set  
ignoreConstructors to true if using solidity >=0.7.0)  
Timelock.sol:111:51: Error: Avoid using low level calls.  
Timelock.sol:120:16: Error: Avoid to make time-based decisions in  
your business logic
```

## HonorDaoChef.sol

```
HonorDaoChef.sol:526:18: Error: Parse error: missing ';' at '{'
```

## HonorDaoStaking.sol

```
HonorDaoStaking.sol:48:18: Error: Parse error: missing ';' at '{'  
HonorDaoStaking.sol:61:18: Error: Parse error: missing ';' at '{'  
HonorDaoStaking.sol:73:18: Error: Parse error: missing ';' at '{'  
HonorDaoStaking.sol:90:18: Error: Parse error: missing ';' at '{'  
HonorDaoStaking.sol:102:18: Error: Parse error: missing ';' at '{'  
HonorDaoStaking.sol:198:18: Error: Parse error: missing ';' at '{'  
HonorDaoStaking.sol:221:18: Error: Parse error: missing ';' at '{'  
HonorDaoStaking.sol:247:18: Error: Parse error: missing ';' at '{'  
HonorDaoStaking.sol:617:18: Error: Parse error: missing ';' at '{'
```

## HonorDaoZapMMSwap.sol

```
HonorDaoZapMMSwap.sol:563:18: Error: Parse error: missing ';' at '{'
```

## Fund.sol

```
Fund.sol:350:18: Error: Parse error: missing ';' at '{'
```

## NFTController.sol

```
NFTController.sol:4:1: Error: Compiler version 0.8.4 does not satisfy the r semver requirement
NFTController.sol:63:28: Error: Avoid using low level calls.
NFTController.sol:137:51: Error: Avoid using low level calls.
NFTController.sol:191:51: Error: Avoid using low level calls.
NFTController.sol:213:17: Error: Avoid using inline assembly. It is acceptable only in rare cases
NFTController.sol:334:5: Error: Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
NFTController.sol:334:20: Error: Code contains empty blocks
NFTController.sol:368:5: Error: Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
NFTController.sol:368:19: Error: Code contains empty blocks
```

## NFTControllerProxy.sol

```
NFTControllerProxy.sol:3:1: Error: Compiler version 0.8.4 does not satisfy the r semver requirement
NFTControllerProxy.sol:59:28: Error: Avoid using low level calls.
NFTControllerProxy.sol:133:51: Error: Avoid using low level calls.
NFTControllerProxy.sol:187:51: Error: Avoid using low level calls.
NFTControllerProxy.sol:209:17: Error: Avoid using inline assembly. It is acceptable only in rare cases
NFTControllerProxy.sol:261:9: Error: Avoid using inline assembly. It is acceptable only in rare cases
NFTControllerProxy.sol:271:9: Error: Avoid using inline assembly. It is acceptable only in rare cases
NFTControllerProxy.sol:281:9: Error: Avoid using inline assembly. It is acceptable only in rare cases
NFTControllerProxy.sol:291:9: Error: Avoid using inline assembly. It is acceptable only in rare cases
NFTControllerProxy.sol:303:9: Error: Avoid using inline assembly. It is acceptable only in rare cases
NFTControllerProxy.sol:365:49: Error: Code contains empty blocks
NFTControllerProxy.sol:541:5: Error: Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
```

```
NFTControllerProxy.sol:558:5: Error: Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
NFTControllerProxy.sol:654:5: Error: Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
NFTControllerProxy.sol:654:114: Error: Code contains empty blocks
```

## DaoFund.sol

```
DaoFund.sol:351:18: Error: Parse error: missing ';' at '{'
```

## DevFund.sol

```
DevFund.sol:350:18: Error: Parse error: missing ';' at '{'
```

## HONORReserve.sol

```
HONORReserve.sol:350:18: Error: Parse error: missing ';' at '{'
```

## MockTreasury.sol

```
MockTreasury.sol:350:18: Error: Parse error: missing ';' at '{'
```

## MockERC20.sol

```
MockERC20.sol:275:18: Error: Parse error: missing ';' at '{'
MockERC20.sol:308:18: Error: Parse error: missing ';' at '{'
MockERC20.sol:357:18: Error: Parse error: missing ';' at '{'
MockERC20.sol:408:22: Error: Parse error: missing ';' at '{'
```

## TreasuryFund.sol

```
TreasuryFund.sol:350:18: Error: Parse error: missing ';' at '{'
```

## MasterOracle.sol

```
MasterOracle.sol:3:1: Error: Compiler version 0.8.4 does not satisfy the r semver requirement
MasterOracle.sol:31:5: Error: Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
MasterOracle.sol:90:5: Error: Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
```

## UniswapPairOracle.sol

```
UniswapPairOracle.sol:483:18: Error: Parse error: missing ';' at '{'
UniswapPairOracle.sol:516:18: Error: Parse error: missing ';' at '{'
UniswapPairOracle.sol:565:18: Error: Parse error: missing ';' at '{'
UniswapPairOracle.sol:616:22: Error: Parse error: missing ';' at '{'
UniswapPairOracle.sol:708:18: Error: Parse error: missing ';' at '{'
UniswapPairOracle.sol:787:18: Error: Parse error: missing ';' at '{'
```

## XToken.sol

```
XToken.sol:276:18: Error: Parse error: missing ';' at '{'
XToken.sol:309:18: Error: Parse error: missing ';' at '{'
XToken.sol:358:18: Error: Parse error: missing ';' at '{'
XToken.sol:409:22: Error: Parse error: missing ';' at '{'
```

## YToken.sol

```
YToken.sol:276:18: Error: Parse error: missing ';' at '{'
YToken.sol:309:18: Error: Parse error: missing ';' at '{'
YToken.sol:358:18: Error: Parse error: missing ';' at '{'
YToken.sol:409:22: Error: Parse error: missing ';' at '{'
```

## HONOR.sol

```
HONOR.sol:276:18: Error: Parse error: missing ';' at '{'
HONOR.sol:309:18: Error: Parse error: missing ';' at '{'
HONOR.sol:358:18: Error: Parse error: missing ';' at '{'
HONOR.sol:409:22: Error: Parse error: missing ';' at '{'
```

## KNIGHTTestToken.sol

```
KNIGHTTestToken.sol:276:18: Error: Parse error: missing ';' at '{'  
KNIGHTTestToken.sol:309:18: Error: Parse error: missing ';' at '{'  
KNIGHTTestToken.sol:358:18: Error: Parse error: missing ';' at '{'  
KNIGHTTestToken.sol:409:22: Error: Parse error: missing ';' at '{'
```

## XNIGHT.sol

```
XNIGHT.sol:276:18: Error: Parse error: missing ';' at '{'  
XNIGHT.sol:309:18: Error: Parse error: missing ';' at '{'  
XNIGHT.sol:358:18: Error: Parse error: missing ';' at '{'  
XNIGHT.sol:409:22: Error: Parse error: missing ';' at '{'
```

## HonorDaoTreasury.sol

```
HonorDaoTreasury.sol:355:18: Error: Parse error: missing ';' at '{'
```

## StratRecollateralize.sol

```
StratRecollateralize.sol:365:18: Error: Parse error: missing ';' at  
'{'
```

## StratReduceReserveLP.sol

```
StratReduceReserveLP.sol:491:18: Error: Parse error: missing ';' at  
'{'
```

## Software analysis result:

These software reported many false positive results and some are informational issues. So, those issues can be safely ignored.



This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

**Email: [audit@EtherAuthority.io](mailto:audit@EtherAuthority.io)**