Ether Authority

www.EtherAuthority.io
audit@etherauthority.io

# SMART CONTRACT

## Security Audit Report

Project: Frontline DAO Protocol
Platform: Polygon Network
Language: Solidity
Date: August 15th, 2022

# Table of contents

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 Frontline DAO to perform the Security audit of the Frontline 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 August 15th, 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

- Frontline 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 Frontline DAO contract inherits the ERC20, SafeERC20, Ownable, ReentrancyGuard, Address, IUniswapV2Router02, IERC20, IERC721, Math, SafeMath, IUniswapV2Pair, Context, Initializable, TransparentUpgradeableProxy, ERC20Burnable 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 Frontline DAO Protocol Smart Contracts** |
|---|---|
| **Platform** | **Polygon / Solidity** |
| **File 1** | Pool.sol |
| **File 1 MD5 Hash** | 856107FFBD51DDC1399BEC38192D0CEB |
| **File 2** | SwapStrategyPOL.sol |
| **File 2 MD5 Hash** | 9CAEB2CD15D2E1C2CECA52DDC4C5FE47 |

| | |
|---|---|
| **File 3** | Timelock.sol |
| **File 3 MD5 Hash** | 94F559046B7CB4335EE0F49341A23DA0 |
| **File 4** | DaoChef.sol |
| **File 4 MD5 Hash** | E12C4E0BDCB405DD0DB61CCF7173ED06 |
| **File 5** | DaoStaking.sol |
| **File 5 MD5 Hash** | 9358FC7E65F92434207DD9F06F01F960 |
| **File 6** | DaoZapMMSwap.sol |
| **File 6 MD5 Hash** | 46D304749327ADF042F8962D64CF8EFC |
| **File 7** | NFTController.sol |
| **File 7 MD5 Hash** | 7B517FFAE5E28C8D3B7020747FFA8659 |
| **File 8** | NFTControllerProxy.sol |
| **File 8 MD5 Hash** | E2E8A433C71CE32907140DCF3F28480D |
| **File 9** | Fund.sol |
| **File 9 MD5 Hash** | 47370A0301A3BBA40747C7FFD8A18E6B |
| **File 10** | DaoFund.sol |
| **File 10 MD5 Hash** | 0DCB7E30D2EE3CBDE722E8D63D87ACF1 |
| **File 11** | DevFund.sol |
| **File 11 MD5 Hash** | A0AC2988CBE65230B36071AA974D05AB |
| **File 12** | Reserve.sol |
| **File 12 MD5 Hash** | 164785B5E9F9181CE4C30A503FC3B0D4 |
| **File 13** | TreasuryFund.sol |
| **File 13 MD5 Hash** | EF21D2BFAFF8EB4877E0B95A99F4B8ED |
| **File 14** | MasterOracle.sol |
| **File 14 MD5 Hash** | 26FFB8A6EB84AABF384A830DB4572C0A |
| **File 15** | UniswapPairOracle.sol |
| **File 15 MD5 Hash** | 37801A23DE6F4571ADD278A4A062C1D5 |
| **File 16** | XToken.sol |

| | |
|---|---|
| **File 16 MD5 Hash** | 83382FC411F2E4462B30C55D6F62A2DD |
| **File 17** | YToken.sol |
| **File 17 MD5 Hash** | FFA9BDAB9AEE9D07DB46CB3A23A34696 |
| **File 18** | LOVE.sol |
| **File 18 MD5 Hash** | 8989CC5C1C75E964DB25F3065208D040 |
| **File 19** | MMFX.sol |
| **File 19  MD5 Hash** | 664C0017F4BF8498B957DB667ED68580 |
| **File 20** | MMOX.sol |
| **File 20 MD5 Hash** | 7B2F2773EA26033423E33F9A1D18FA02 |
| **File 21** | DaoTreasury.sol |
| **File 21 MD5 Hash** | B4AEF5736647E39B4AEC058D7AF7A989 |
| **File 22** | MMUSD.sol |
| **File 22 MD5 Hash** | 9CE7E628B8A5C3D365CF307F3C64AD15 |
| **File 23** | StratRecollateralize.sol |
| **File 23 MD5 Hash** | 8825619BF2C90BD2337916B8B6CB90B2 |
| **File 24** | StratReduceReserveLP.sol |
| **File 24 MD5 Hash** | EE1CAC2A02D76BCD4158A836C13CE2DB |
| **Audit Date** | August 15th,2022 |

## Claimed Smart Contract Features

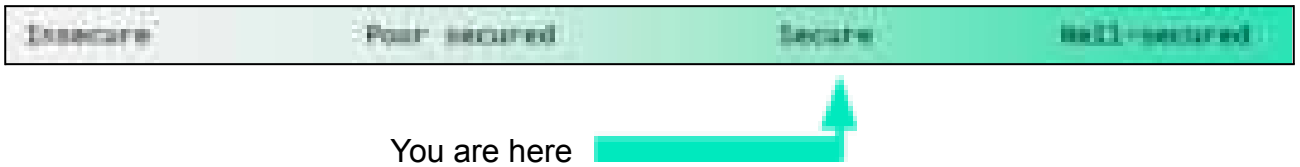
| Claimed Feature Detail | Our Observation |
|---|---|
| **File 1 Pool.sol**<br>• Refresh Cooldown: 1 hour<br>• Ratio StepUp: 0.2%<br>• Ratio StepDown: 0.2%<br>• Price Target: 1 MMF<br>• Price Band: 0.004<br>• Minimum Collateral Ratio: 9,50,000<br>• YToken Slippage: 20%<br>• Redemption Fee: 0.5%<br>• Redemption Fee Maximum: 0.9%<br>• Minting Fee: 0.3%<br>• Minting Fee Maximum: 0.5% | **YES, This is valid.** |
| **File 2 SwapStrategyPOL.sol**<br>• Swap Slippage: 20% | **YES, This is valid.** |
| **File 3 Timelock.sol**<br>• Grace Period: 14 Days<br>• Minimum Delay: 12 Hours<br>• Maximum Delay: 30 Days | **YES, This is valid.** |
| **File 4 DaoChef.sol**<br>• Maximum reward: 10 Tokens Per Second<br>• NFT Boost Rate: 100 | **YES, This is valid.** |
| **File 5 DaoStaking.sol**<br>• Rewards Duration: 1 week<br>• Lock Duration: 4 weeks<br>• Team Reward Percent: 20% | **YES, This is valid.** |
| **File 6 DaoZapMMSwap.sol**<br>DaoZap is a ZapperFi's simplified version of zapper | **YES, This is valid.** |

| | |
|---|---|
| contract which will:<br>1. use ETH to swap to target token<br>2. make LP between ETH and target token<br>3. add into DaoChef farm | |
| **File 7 Fund.sol**<br>• Fund has functions like: allocation, transfer, etc. | **YES, This is valid.** |
| **File 8 DaoFund.sol**<br>• Allocation: 10%<br>• Vesting Duration: 3 Years | **YES, This is valid.** |
| **File 9 DevFund.sol**<br>• Allocation: 10%<br>• Vesting Duration: 2 Years | **YES, This is valid.** |
| **File 10 Reserve.sol**<br>• Reserve has functions like: initialize, etc. | **YES, This is valid.** |
| **File 11 TreasuryFund.sol**<br>• Allocation: 10%<br>• Vesting Duration: 3 Years | **YES, This is valid.** |
| **File 12 MasterOracle.sol**<br>• MasterOracle has functions like getYTokenTWAP, etc. | **YES, This is valid.** |
| **File 13 UniswapPairOracle.sol**<br>• Period: 60-minute Twap (Time-weighted Average Price)<br>• Maximum Period: 48 Hours<br>• Minimum Period: 10 Minutes<br>• Leniency: 12 Hours | **YES, This is valid.** |
| **File 14 XToken.sol**<br>• XToken has functions like: mint, setMinter, etc. | **YES, This is valid.** |
| **File 15 YToken.sol** | **YES, This is valid.** |

| | |
|---|---|
| • YToken has functions like: burn, etc. | |
| **File 16 LOVE.sol**<br>• Total Supply: 30 Million | **YES, This is valid.** |
| **File 17 MMFX.sol**<br>• Genesis Supply: 100 will be minted at genesis for liq pool seeding | **YES, This is valid.** |
| **File 18 MMOX.sol**<br>• Genesis Supply: 100 will be minted at genesis for liq pool seeding | **YES, This is valid.** |
| **File 19 MMUSD.sol**<br>• Genesis Supply: 100 will be minted at genesis for liq pool seeding | **YES, This is valid.** |
| **File 20 DaoTreasury.sol**<br>• DaoTreasury has functions like: balanceOf, requestFund, etc. | **YES, This is valid.** |
| **File 21 StratRecollateralize.sol**<br>• StratRecollateralize has functions like: recollateralize. | **YES, This is valid.** |
| **File 22 StratReduceReserveLP.sol**<br>• StratReduceReserveLPhas functions like: reduceReserve, swap. | **YES, This is valid.** |
| **File 23 NFTController.sol**<br>• NFTController has functions like: setBoostRate, setWhitelist, etc. | **YES, This is valid.** |
| **File 24 NFTControllerProxy.sol**<br>• NFTControllerProxy has access to the TransparentUpgradeableProxy contract. | **YES, This is valid.** |

# 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 | Passed |
| 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 24 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 Frontline 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 Frontline DAO Protocol.

The Frontline 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 a Frontline 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.

## 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 | receive | external | Passed | No Issue |
| 15 | mint | external | Passed | No Issue |
| 16 | redeem | external | Passed | No Issue |
| 17 | collect | external | Passed | No Issue |
| 18 | recollateralize | external | Passed | No Issue |
| 19 | checkPriceFluctuation | internal | Passed | No Issue |
| 20 | toggle | write | access only Owner | No Issue |
| 21 | setCollateralRatioOptions | write | access only Owner | No Issue |
| 22 | toggleCollateralRatio | write | access only Owner | No Issue |
| 23 | setFees | write | access only Owner | No Issue |
| 24 | setMinCollateralRatio | external | access only Owner | No Issue |
| 25 | reduceExcessCollateral | external | access only Owner | No Issue |
| 26 | setSwapStrategy | external | access only Owner | No Issue |
| 27 | setOracle | external | access only Owner | No Issue |
| 28 | setYTokenSlippage | external | access only Owner | No Issue |
| 29 | setTreasury | external | Passed | No Issue |
| 30 | 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 |

## DaoChef.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 |

## DaoStaking.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** | addReward | write | Function input parameters lack of check | Refer Audit Findings |
| **8** | approveRewardDistributor | external | access only Owner | No Issue |
| **9** | _rewardPerToken | internal | Passed | No Issue |
| **10** | _earned | internal | Passed | No Issue |
| **11** | lastTimeRewardApplicable | read | Passed | No Issue |
| **12** | rewardPerToken | external | 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** | lockedBalances | external | Passed | No Issue |
| **19** | withdrawableBalance | read | Passed | No Issue |
| **20** | stake | external | Passed | No Issue |
| **21** | mint | external | Function input parameters lack of check | Refer Audit Findings |
| **22** | withdraw | write | Passed | No Issue |
| **23** | getReward | write | Passed | No Issue |
| **24** | emergencyWithdraw | external | Critical operation lacks event log | Refer Audit Findings |
| **25** | withdrawExpiredLocks | external | Passed | No Issue |
| **26** | _notifyReward | internal | Passed | No Issue |
| **27** | notifyRewardAmount | external | Passed | No Issue |

| | | | | |
|---|---|---|---|---|
| **28** | recoverERC20 | external | access only Owner | No Issue |
| **29** | setTeamWalletAddress | external | Passed | No Issue |
| **30** | setTeamRewardPercent | external | Passed | No Issue |
| **31** | updateReward | modifier | Passed | No Issue |
| **32** | receive | external | Passed | No Issue |

## DaoZapMMSwap.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 | access by initializer | 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 |

## Reserve.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 | access by initializer | No Issue |
| **13** | setRewarder | external | Passed | No Issue |
| **14** | setPool | external | access only Owner | No Issue |
| **15** | removePool | external | access only Owner | No Issue |

| | | | | |
|---|---|---|---|---|
| **16** | 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 |

## 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 | onlyMinter | modifier | Passed | No Issue |
| 3 | setMinter | external | access only Owner | No Issue |
| 4 | removeMinter | external | access only Owner | No Issue |
| 5 | mint | external | Unlimited Minting | Refer Audit Findings |
| 6 | burn | write | Passed | No Issue |
| 7 | burnFrom | write | Passed | No Issue |
| 8 | owner | read | Passed | No Issue |
| 9 | onlyOwner | modifier | Passed | No Issue |
| 10 | renounceOwnership | write | access only Owner | No Issue |
| 11 | transferOwnership | write | access only Owner | No Issue |
| 12 | _transferOwnership | internal | Passed | No Issue |

## 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 |

## LOVE.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 | excludeFromWhitlist | write | Passed | No Issue |
| 5 | _transfer | internal | Passed | No Issue |

## MMFX.sol

**Functions**

| Sl. | Functions | Type | Observation | Conclusion |
|---|---|---|---|---|
| 1 | constructor | write | Passed | No Issue |
| 2 | onlyMinter | modifier | Passed | No Issue |
| 3 | setMinter | external | access only Owner | No Issue |
| 4 | removeMinter | external | access only Owner | No Issue |
| 5 | mint | external | access only Minter | No Issue |
| 6 | OpenTrade | external | Passed | No Issue |
| 7 | includeToWhitelist | write | Passed | No Issue |
| 8 | excludeFromWhitlist | write | Passed | No Issue |
| 9 | _transfer | internal | Passed | No Issue |

## MMOX.sol

**Functions**

| Sl. | Functions | Type | Observation | Conclusion |
|---|---|---|---|---|
| 1 | constructor | write | Passed | No Issue |
| 2 | onlyMinter | modifier | Passed | No Issue |
| 3 | setMinter | external | access only Owner | No Issue |
| 4 | removeMinter | external | access only Owner | No Issue |
| 5 | mint | external | access only Minter | No Issue |
| 6 | OpenTrade | external | Passed | No Issue |
| 7 | includeToWhitelist | write | Passed | No Issue |
| 8 | excludeFromWhitlist | write | Passed | No Issue |
| 9 | _transfer | internal | Passed | No Issue |

## MMUSD.sol

**Functions**

| Sl. | Functions | Type | Observation | Conclusion |
|---|---|---|---|---|
| 1 | constructor | write | Passed | No Issue |
| 2 | onlyMinter | modifier | Passed | No Issue |
| 3 | setMinter | external | access only Owner | No Issue |
| 4 | removeMinter | external | access only Owner | No Issue |
| 5 | mint | external | access only Minter | No Issue |
| 6 | OpenTrade | external | Passed | No Issue |
| 7 | includeToWhitelist | write | Passed | No Issue |
| 8 | excludeFromWhitlist | write | Passed | No Issue |
| 9 | _transfer | internal | Passed | No Issue |

## DaoTreasury.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 | r | 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 |
| 11 | allocateFee | external | access only Owner | No Issue |

## StratRecollateralize.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 | recollateralize | external | access only Owner | No Issue |
| 8 | receive | external | 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 |
|---|---|
| Critical | Critical vulnerabilities are usually straightforward to exploit and can lead to token loss etc. |
| High | High-level vulnerabilities are difficult to exploit; however, they also have significant impact on smart contract execution, e.g. public access to crucial |
| Medium | Medium-level vulnerabilities are important to fix; however, they can't lead to tokens lose |
| Low | Low-level vulnerabilities are mostly related to outdated, unused etc. code snippets, that can't have significant impact on execution |
| Lowest / Code Style / Best Practice | 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:**- DaoStaking.sol**

Missing event log for: emergencyWithdraw

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

(2) Function input parameters lack of check: **- DaoStaking.sol**

Variable validation is not performed in the functions below:

- 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.

## 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 yTokenSlipage.
- setTreasury: Pool owner can set the address of the Treasury.
- cleanDust: SwapStrategyPOL owner can clean dust.
- changeSlippage: SwapStrategyPOL owner can change slippage value.
- add: DaoChef owner can add a new LP to the pool.
- set: DaoChef owner can update the given pool's reward allocation point and `IRewarder` contract.
- setRewardPerSecond: DaoChef owner can set the reward per second to be distributed.
- setRewardMinter: DaoChef owner can set the address of rewardMinter.
- setNftController: DaoChef owner can set NFT controller address.
- setNftBoostRate:  DaoChef owner can set NFT Boost Rate value.
- addReward: DaoStaking owner can add a new reward token to be distributed to stakers.
- approveRewardDistributor: DaoStaking owner can modify approval for an address to call notifyRewardAmount.
- recoverERC20: DaoStaking  owner can be added to support recovering LP Rewards from other systems such as BAL to be distributed to holders.

- setTeamWalletAddress: DaoStaking  owner can set the address of the team wallet.
- setTeamRewardPercent: DaoStaking  owner can set percent of team reward.
- addZap:  DaoZapMMSwap owner can add new zap configuration.
- removeZap: DaoZapMMSwap 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.
- setPool: Reserve owner can set pool address.
- removePool: Reserve owner can remove pool address.
- setPeriod: UniswapPairOracle owner can set maximum and minimum period.
- removeMinter: XToken minter can remove minter address.
- setMinter: XToken minter can set minter for XToken.
- mint: XToken minter can mint new XToken.
- OpenTrade: LOVE  owners can trade openly.
- includeToWhitelist: LOVE owner can include address  to whitelist.
- excludeFromWhitlist: LOVE owner can exclude address  to whitelist.
- OpenTrade: MMFX  owners can trade openly.
- includeToWhitelist: MMFX owner can include address  to whitelist.
- excludeFromWhitlist:  MMFX owner can exclude address  to whitelist.
- OpenTrade: MMOX owners can trade openly.
- includeToWhitelist: MMOX owner can include address  to whitelist.
- excludeFromWhitlist: MMOX owner can exclude address  to whitelist.
- OpenTrade: MMUSD owners can trade openly.
- includeToWhitelist: MMUSD owner can include address  to whitelist.
- excludeFromWhitlist:  MMUSD owner can exclude address  to whitelist.
- addStrategy: DaoTreasury owner can add new strategy.
- removeStrategy: DaoTreasury owner can  remove current strategy.
- allocateFee: DaoTreasury  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.

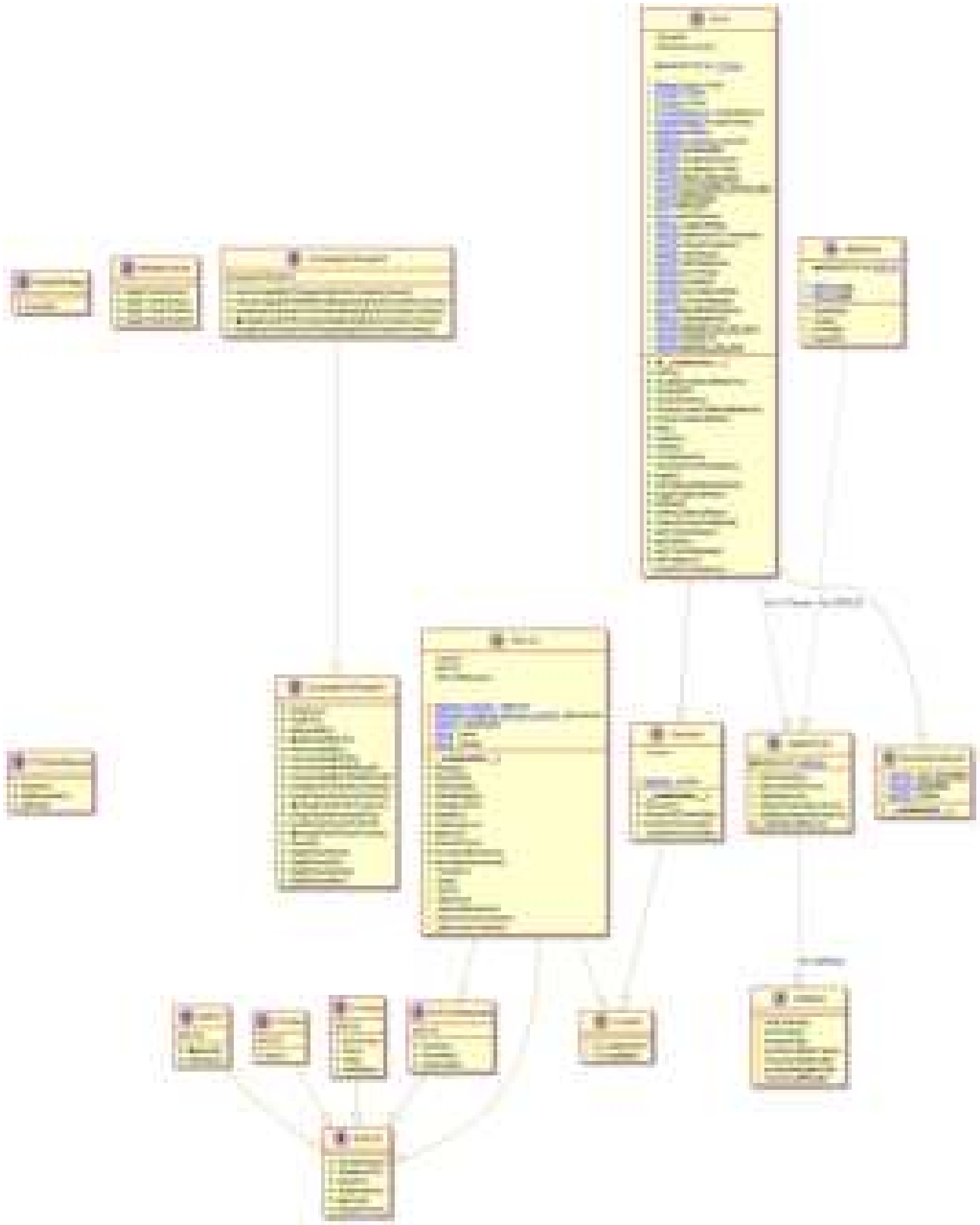
## Technical Disclaimer

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.

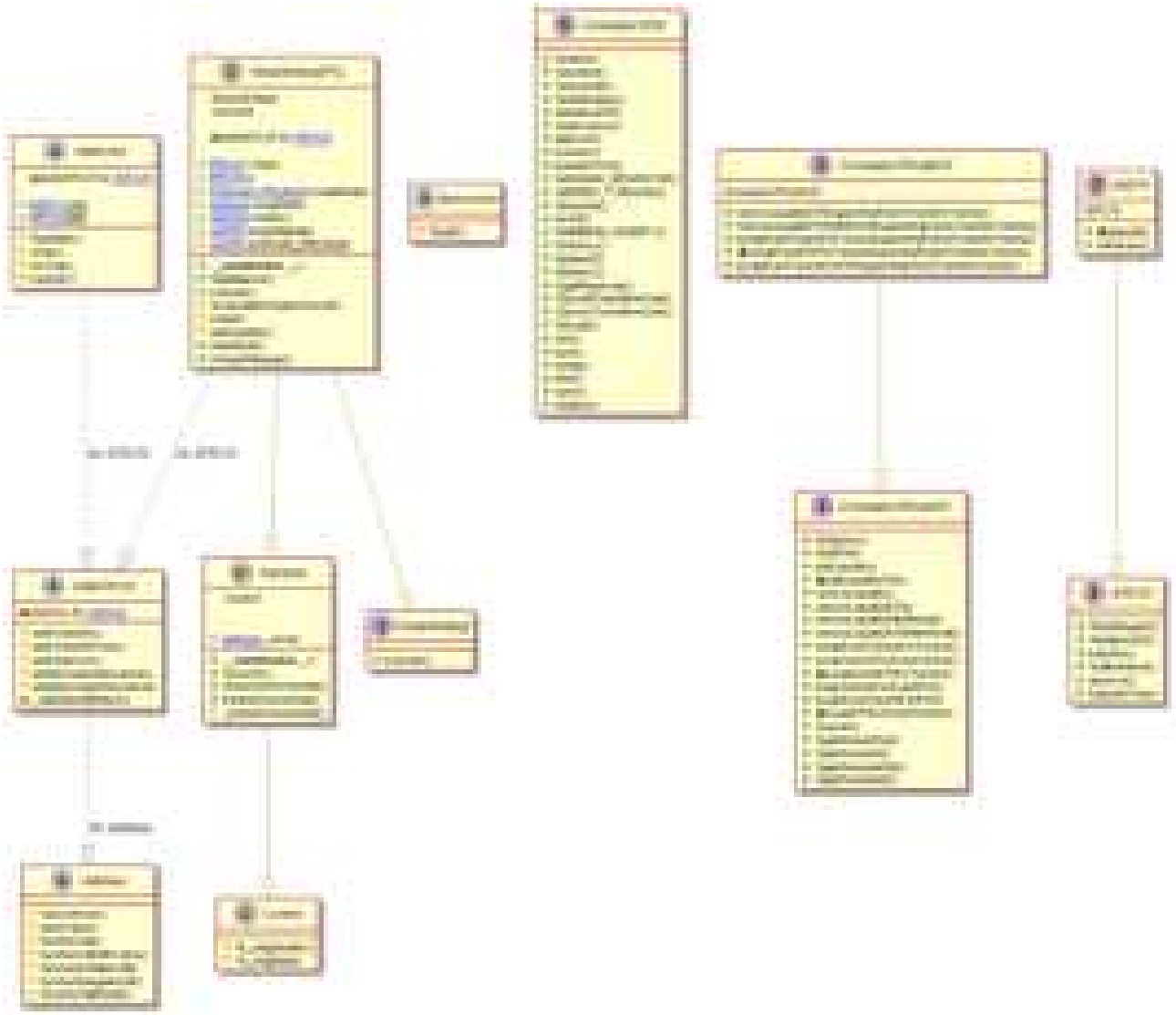
# Appendix

## Code Flow Diagram - Frontline  DAO Protocol

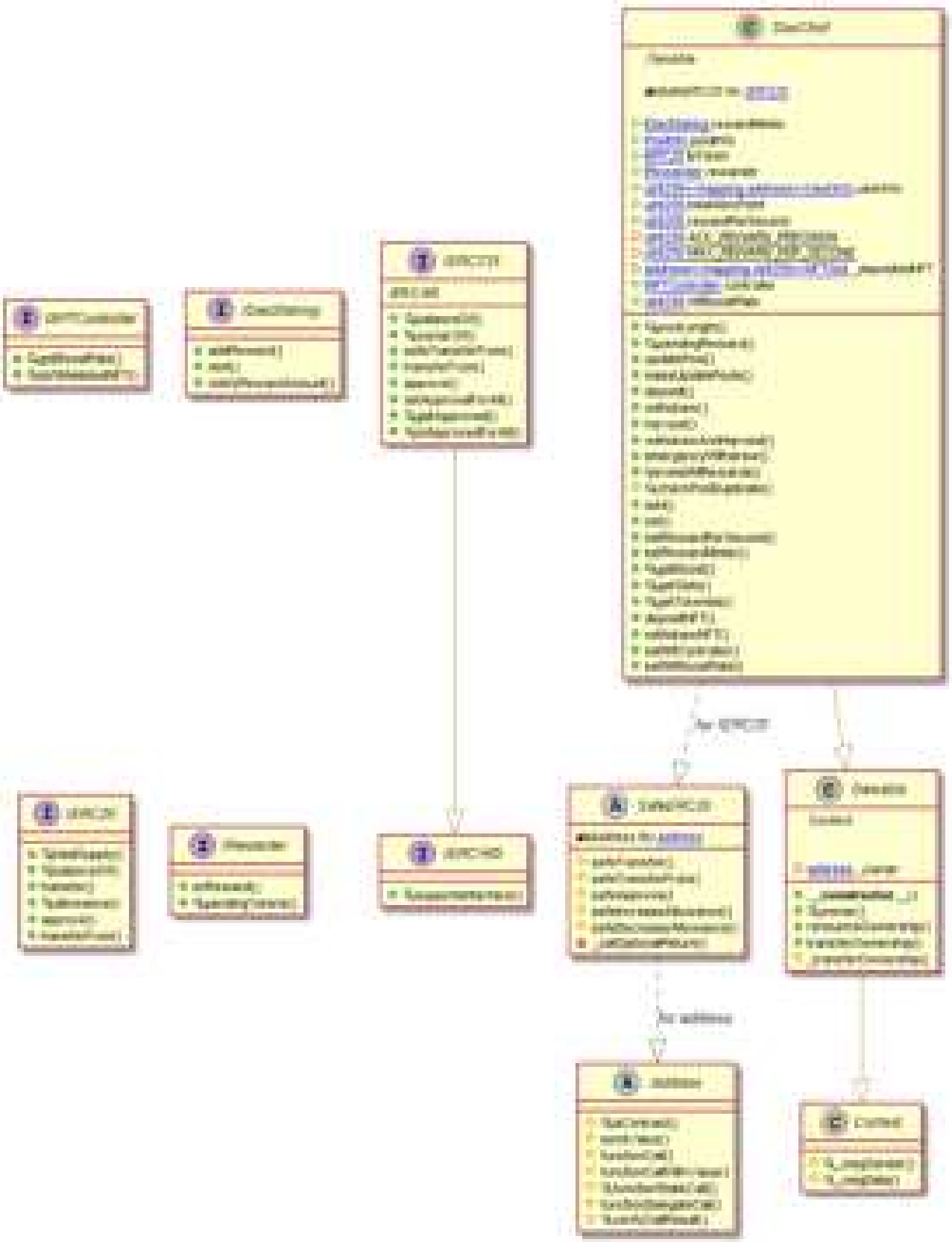
### Pool Diagram



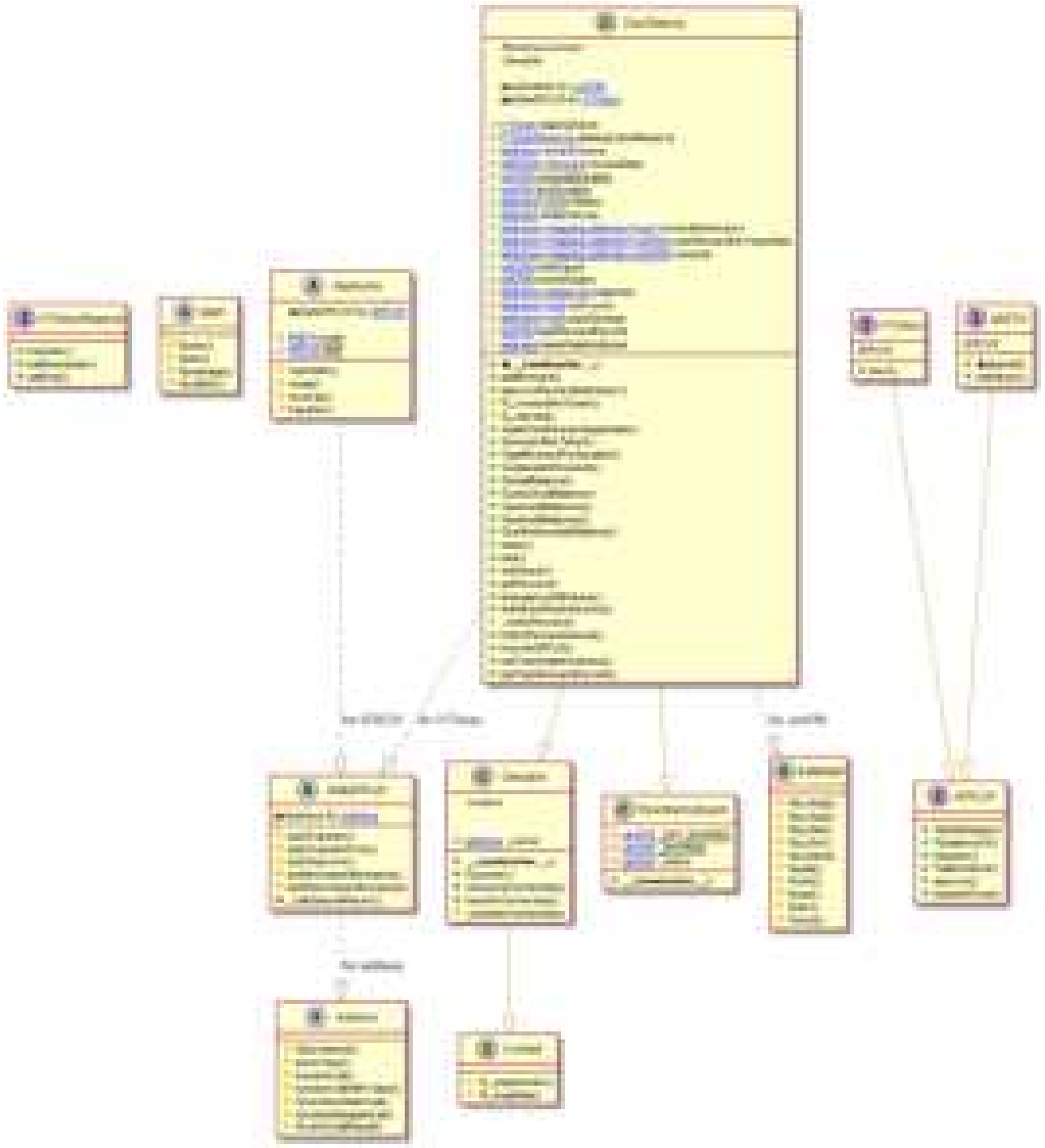
## SwapStrategyPOL Diagram



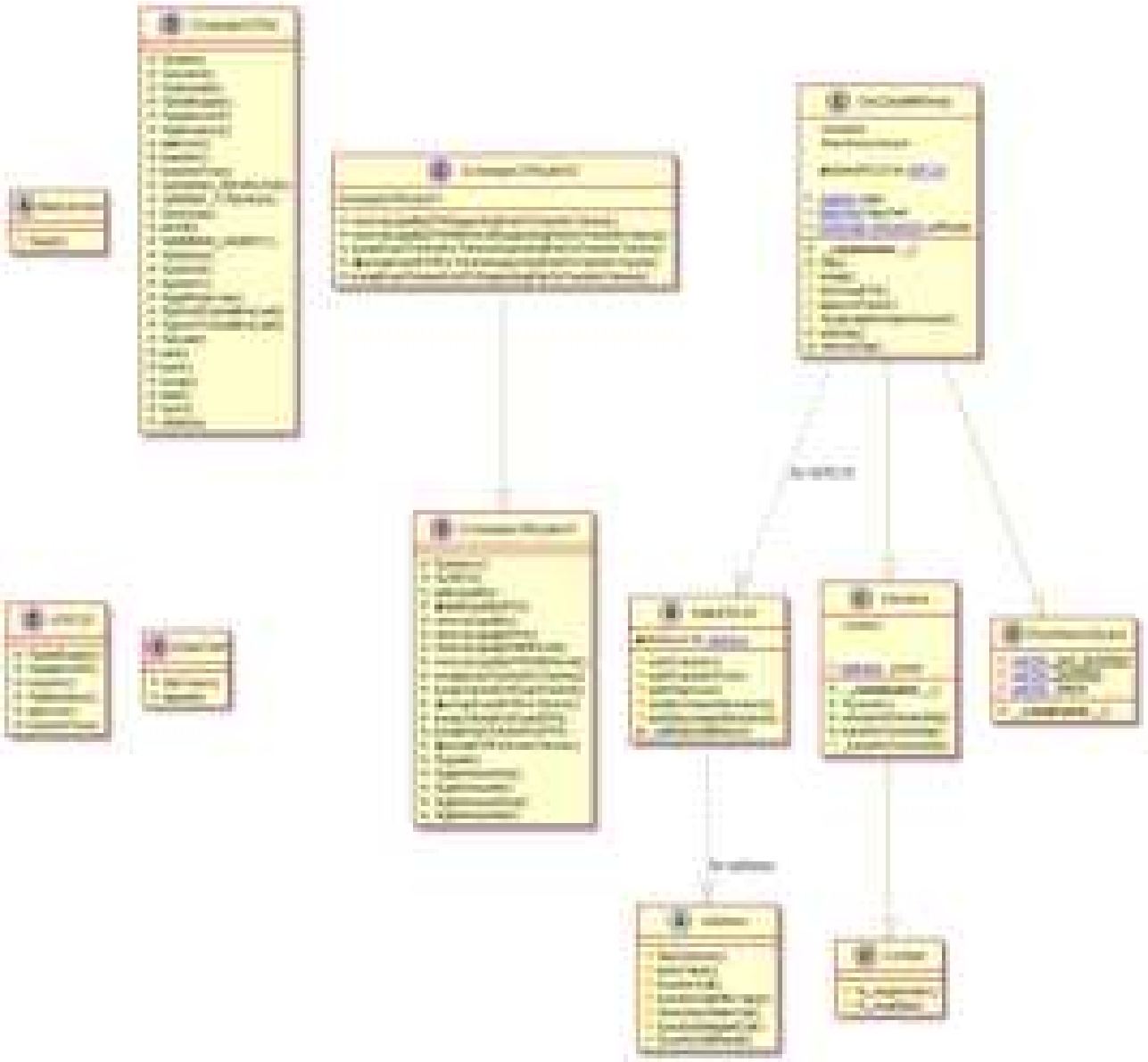
# DaoChef Diagram



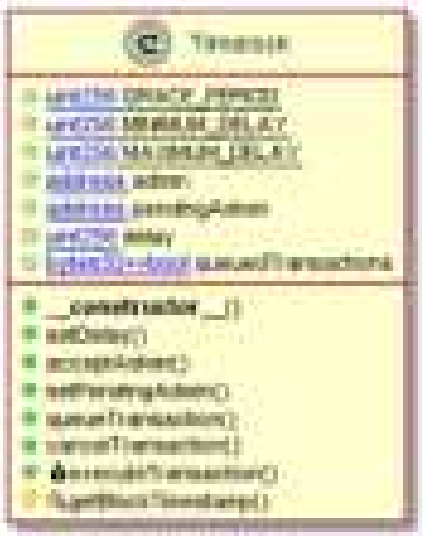
# DaoStaking Diagram



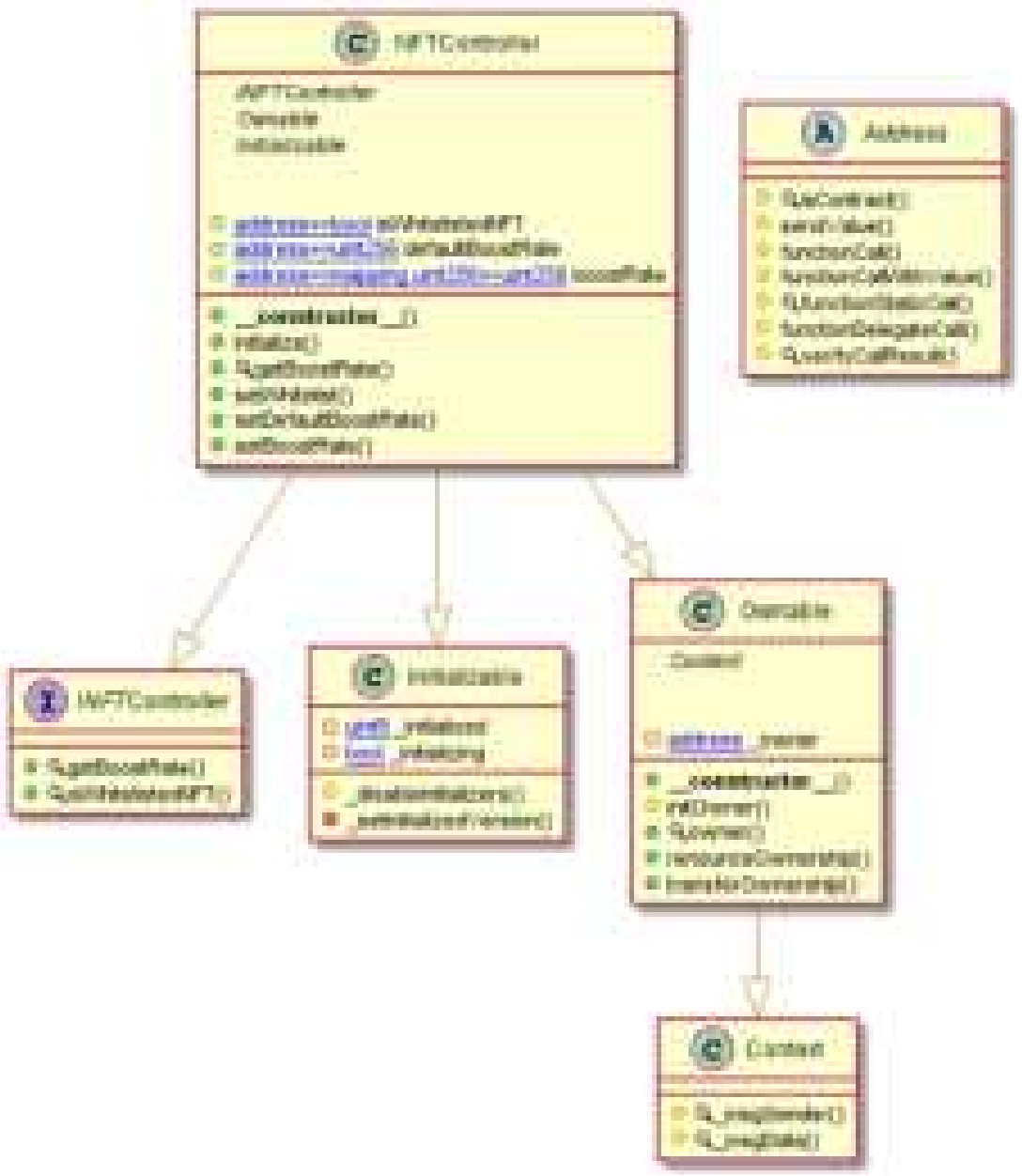
## DaoZapMMSwap Diagram



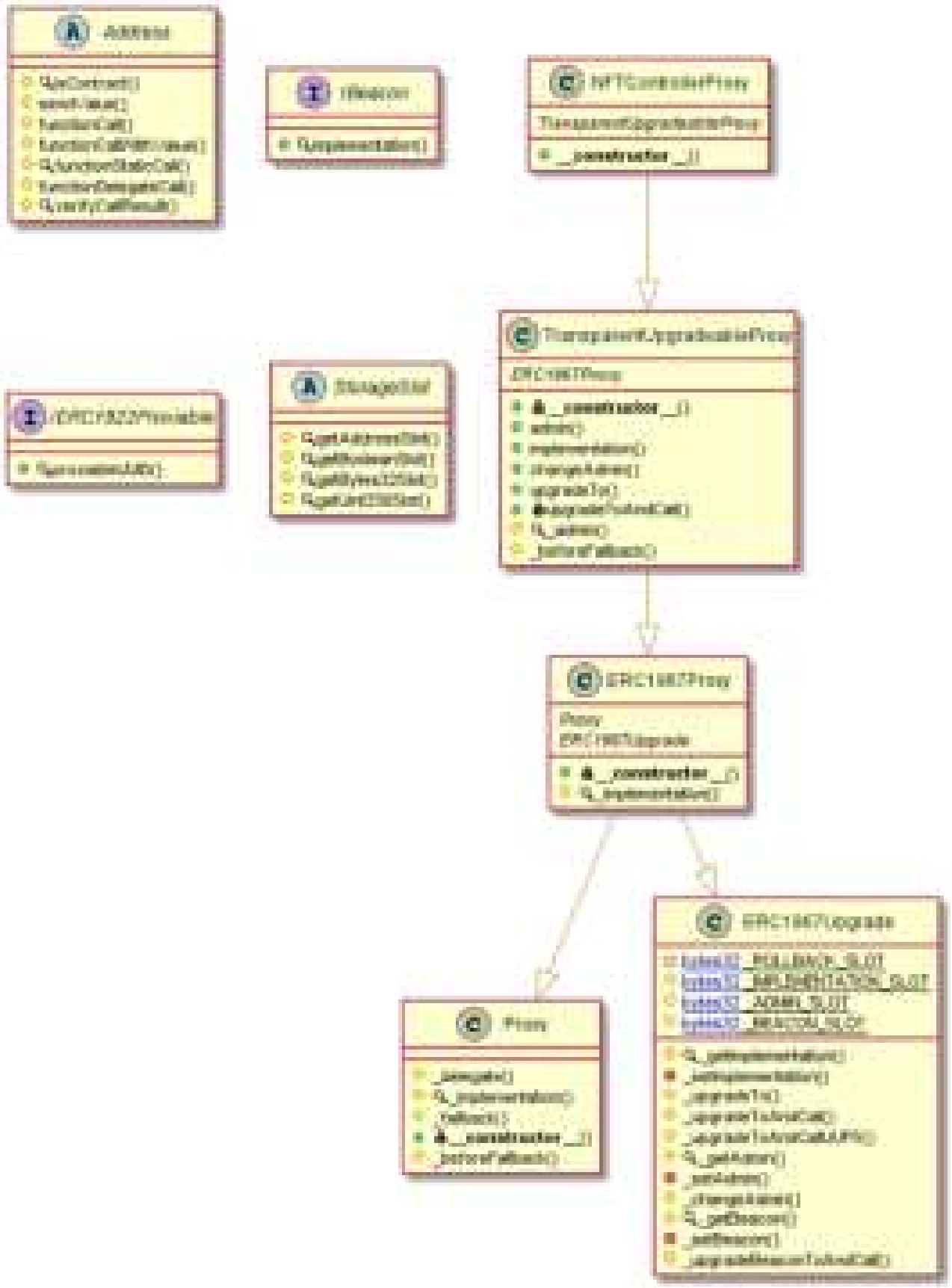
## Timelock Diagram



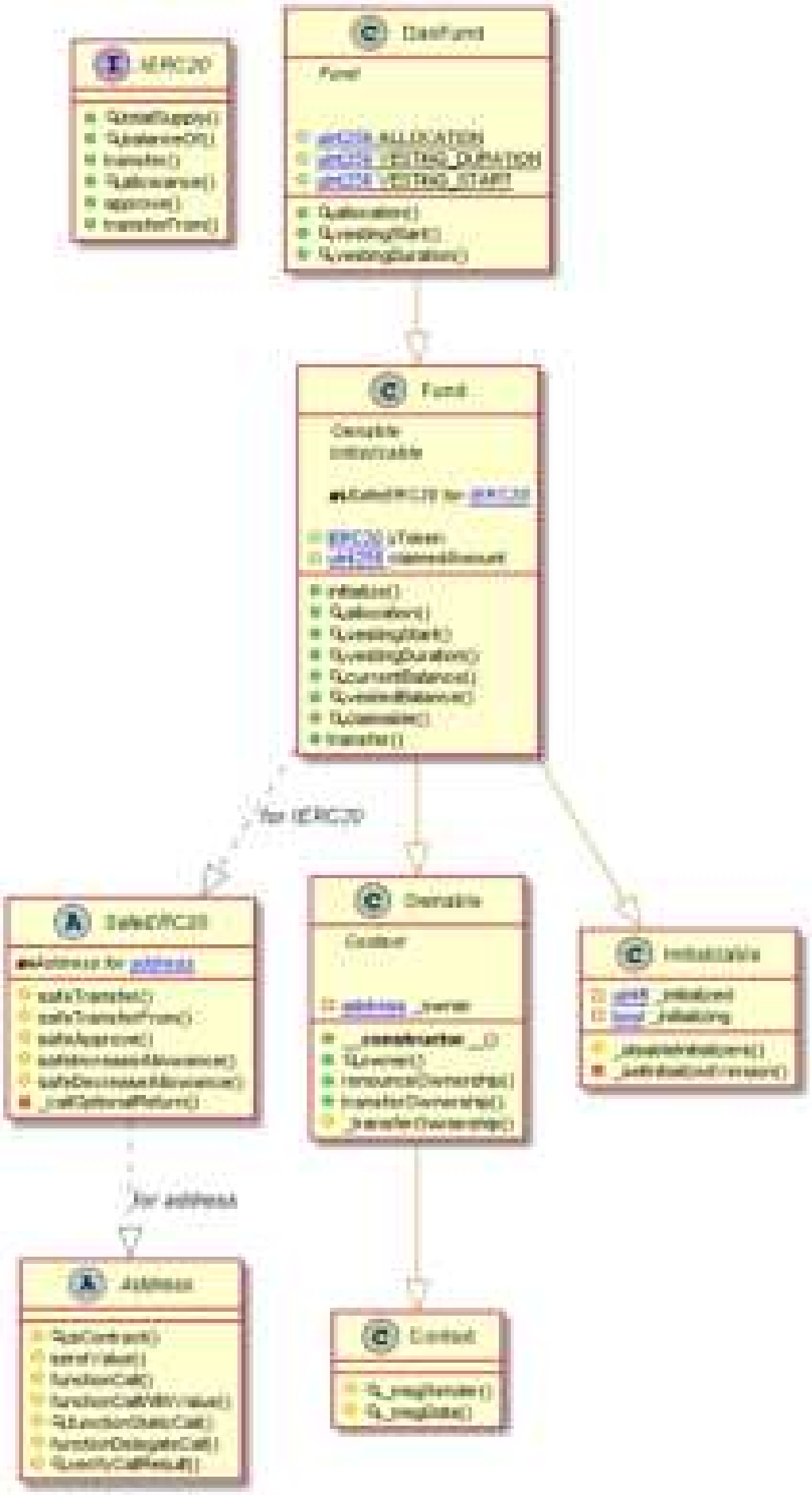
## NFTController Diagram



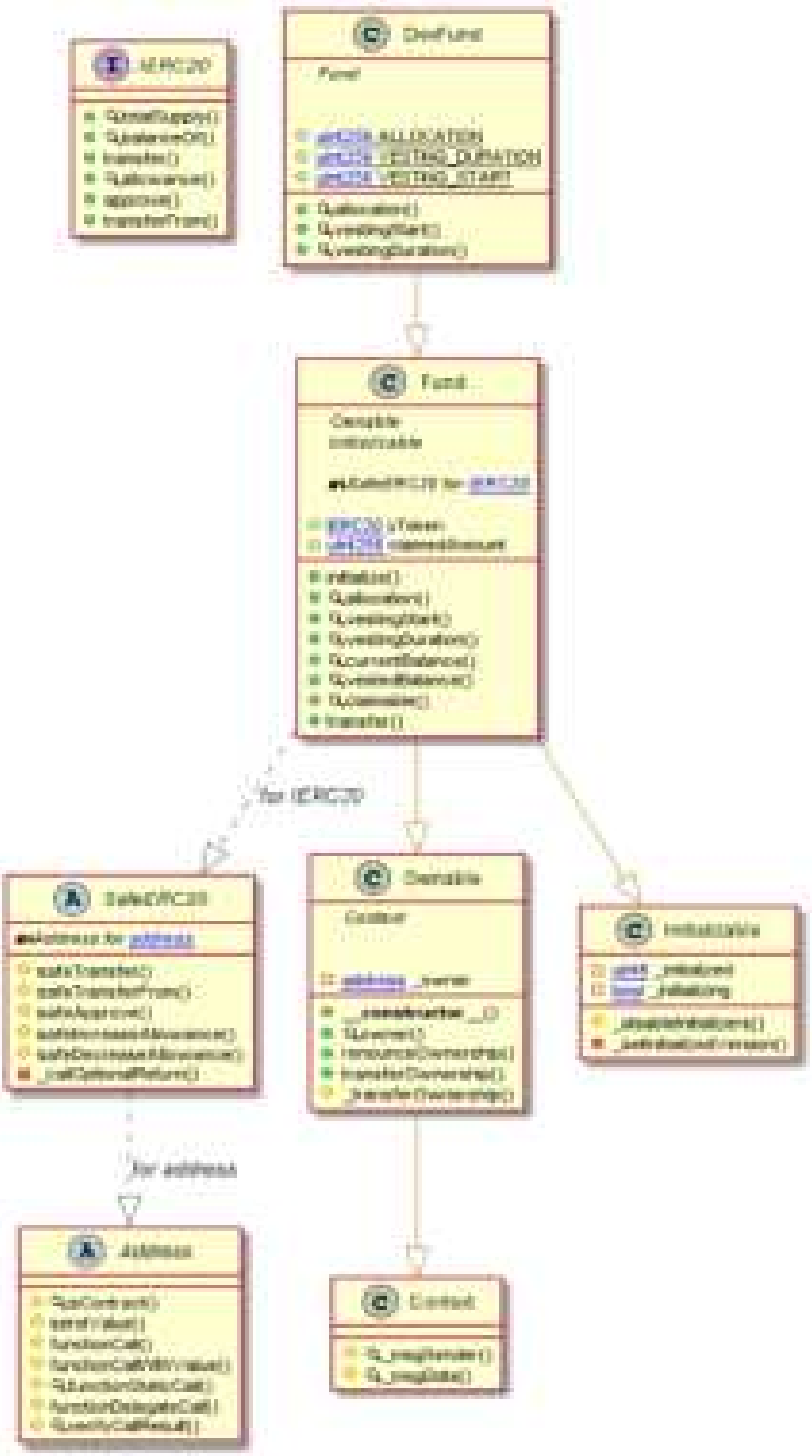
## NFTControllerProxy Diagram



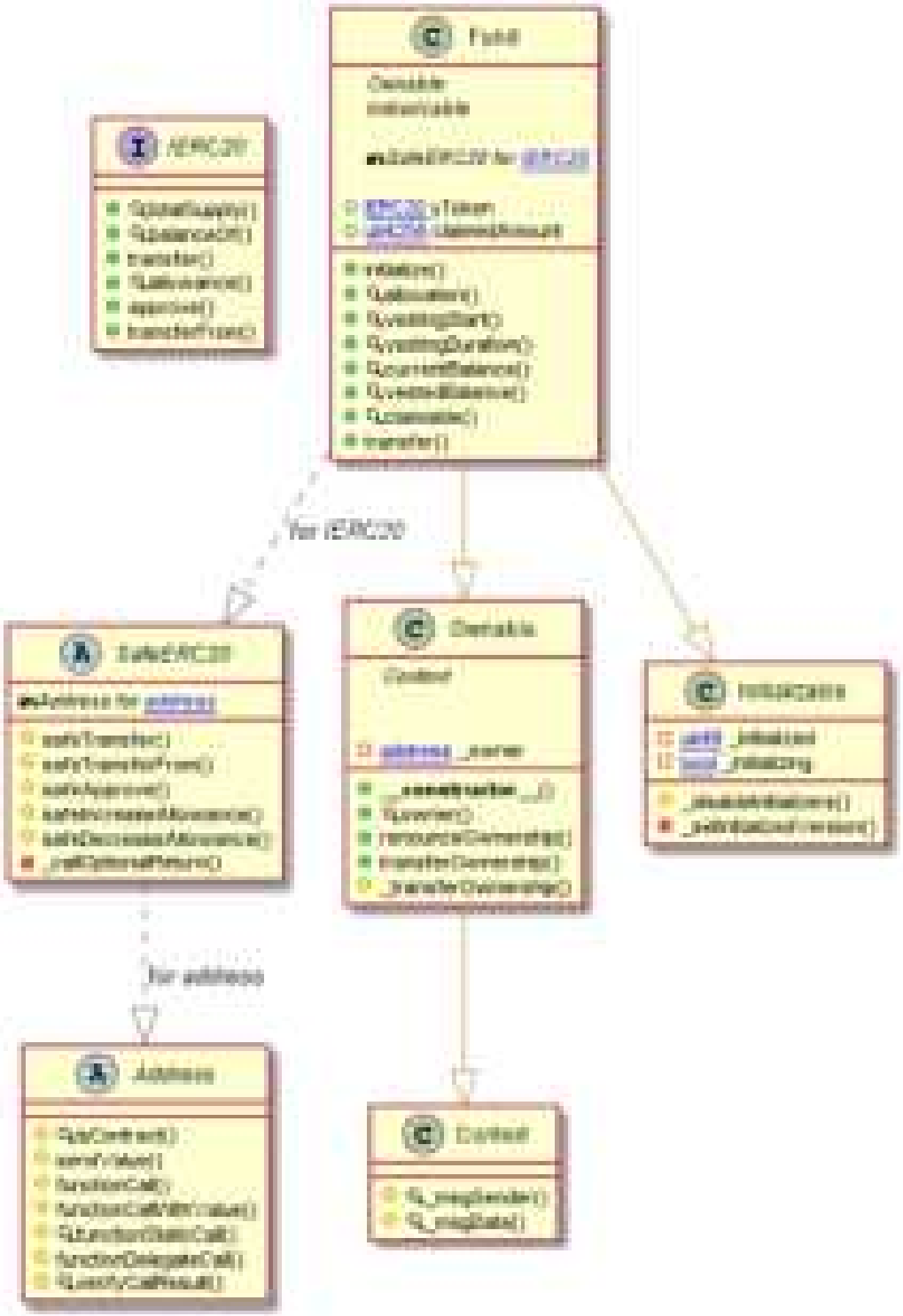
## DaoFund Diagram



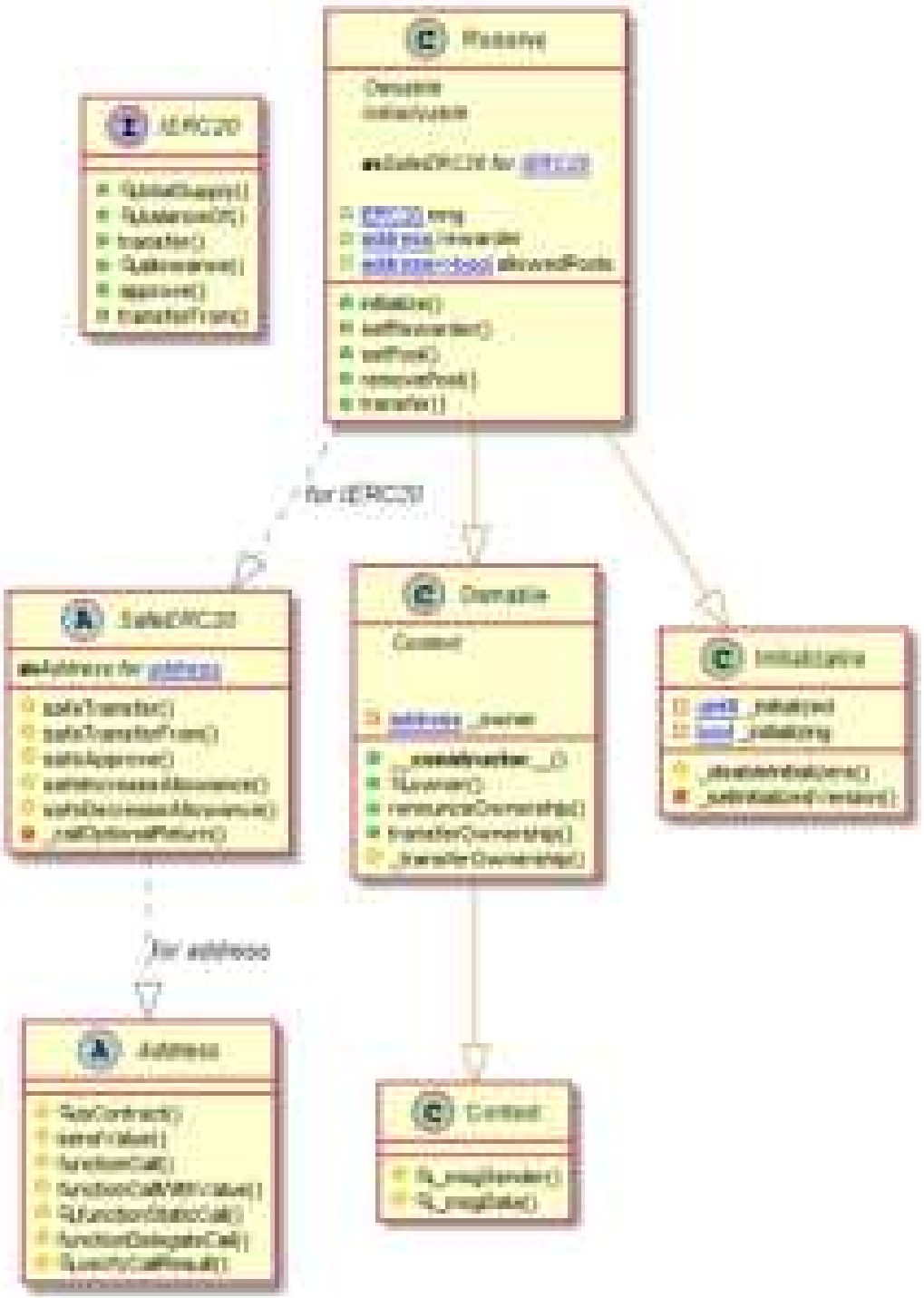
# DevFund Diagram



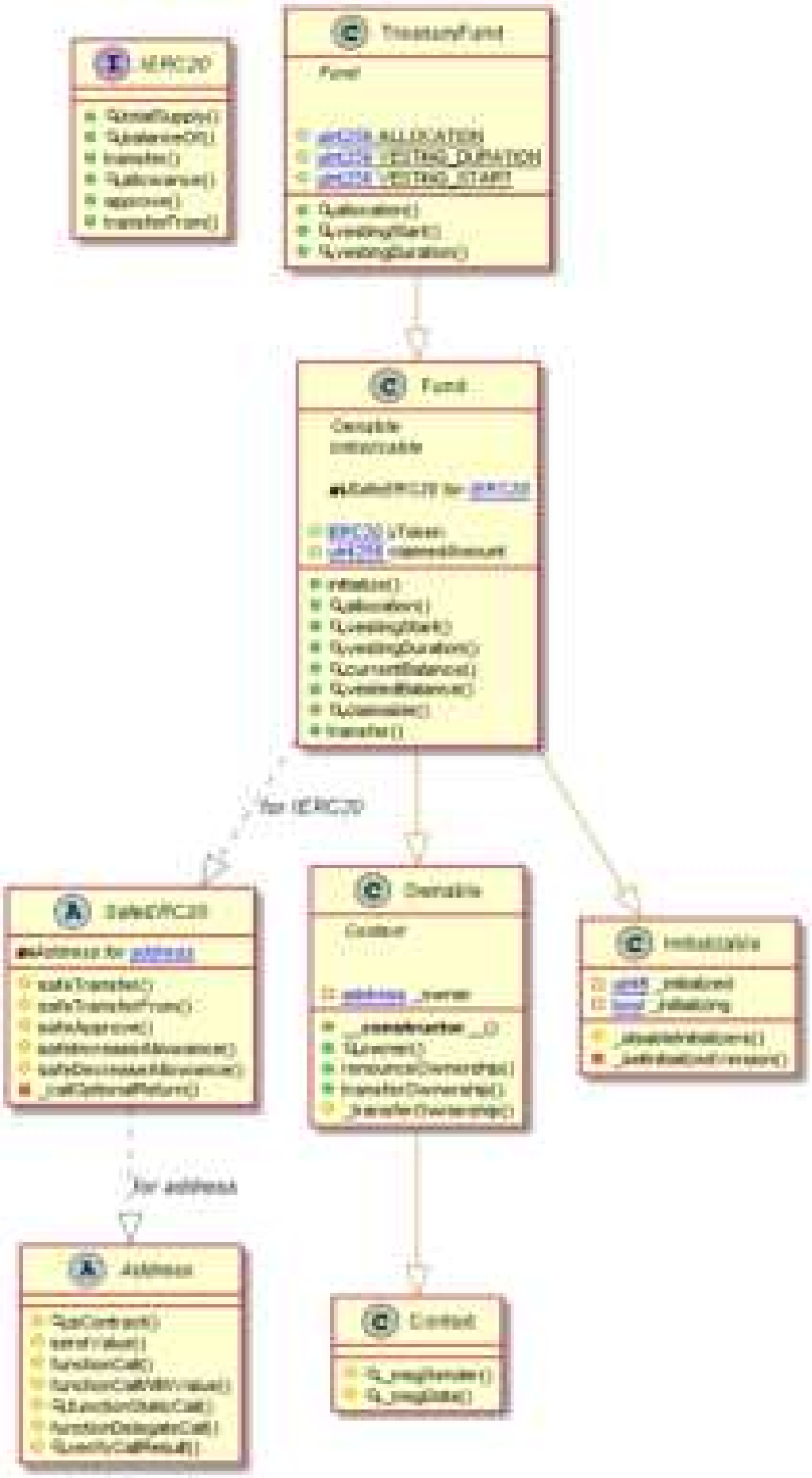
## Fund Diagram



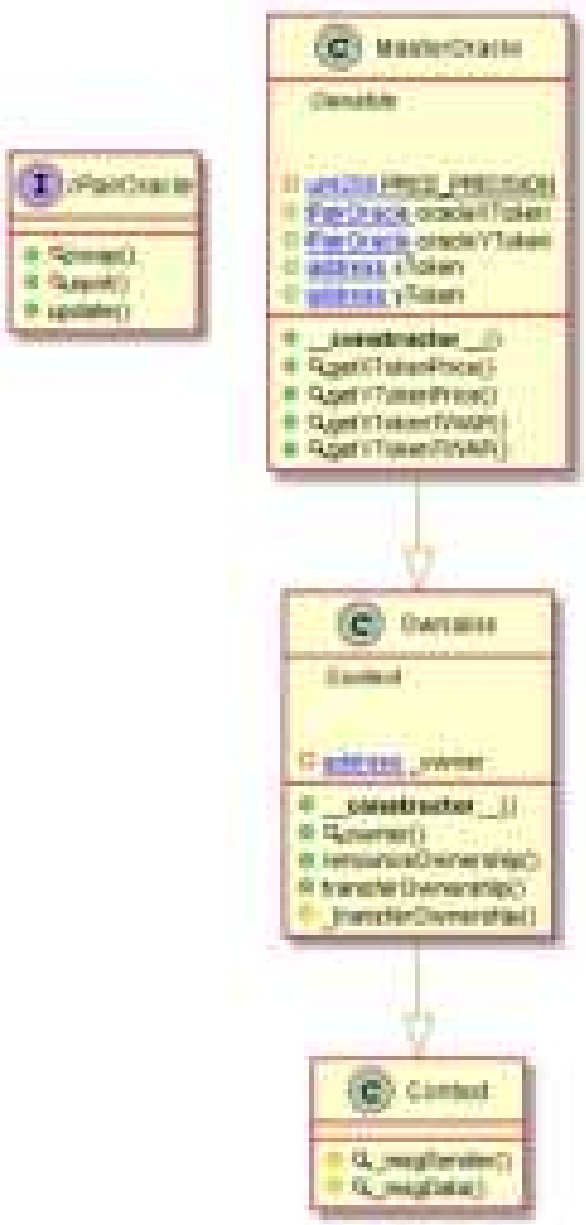
## Reserve Diagram



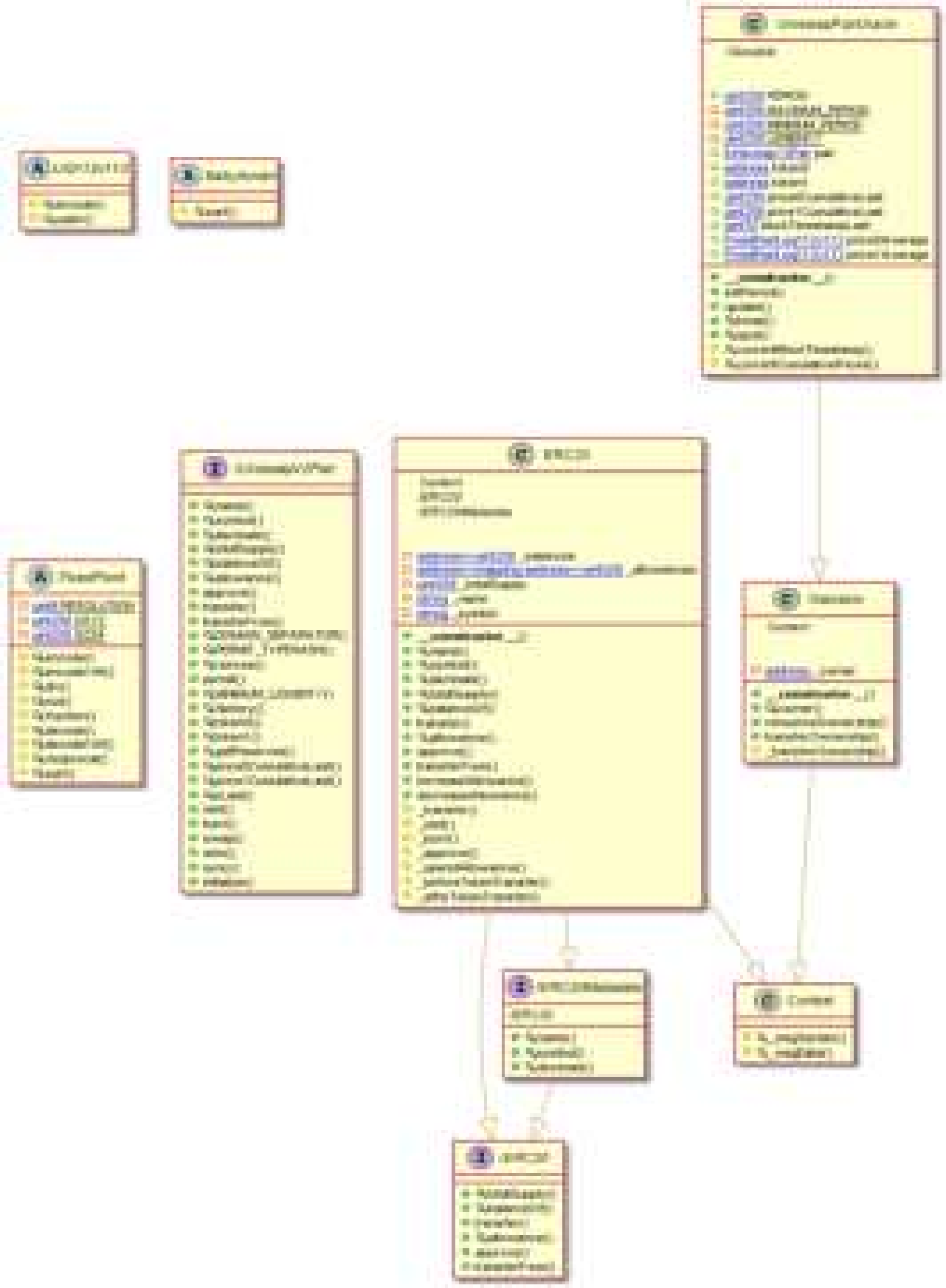
## TreasuryFund Diagram



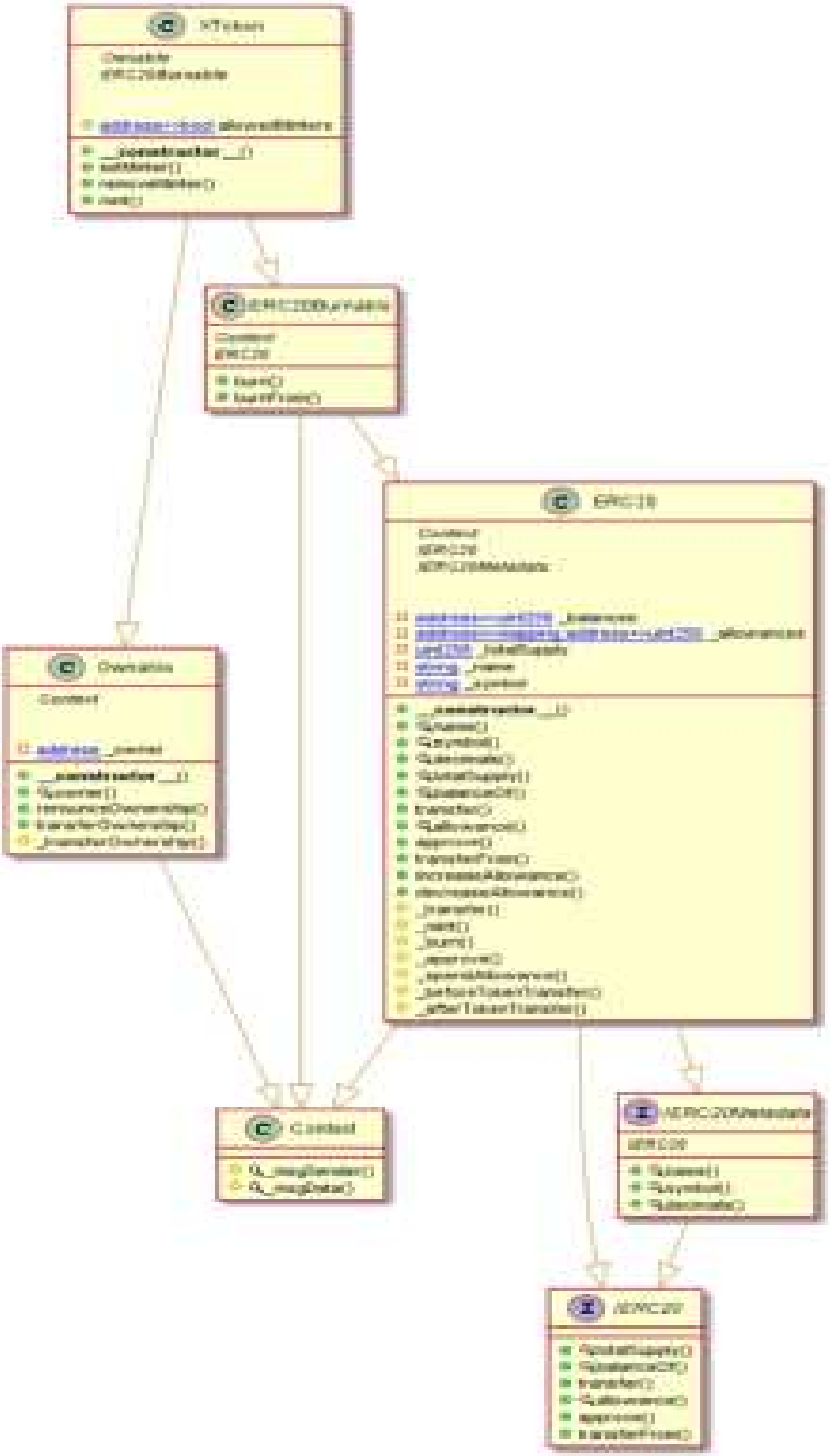
## MasterOracle Diagram



# UniswapPairOracle Diagram



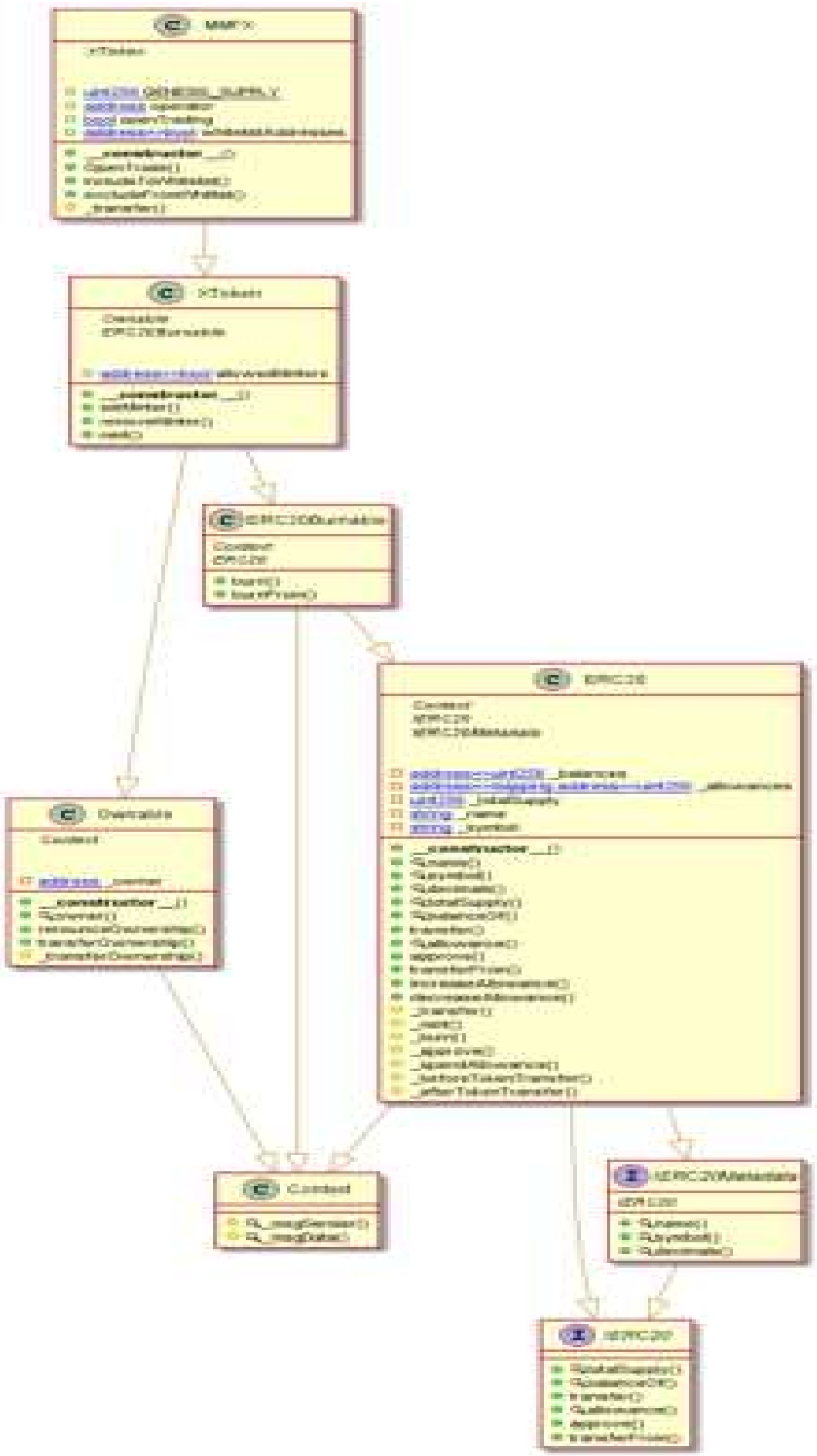
# XToken Diagram



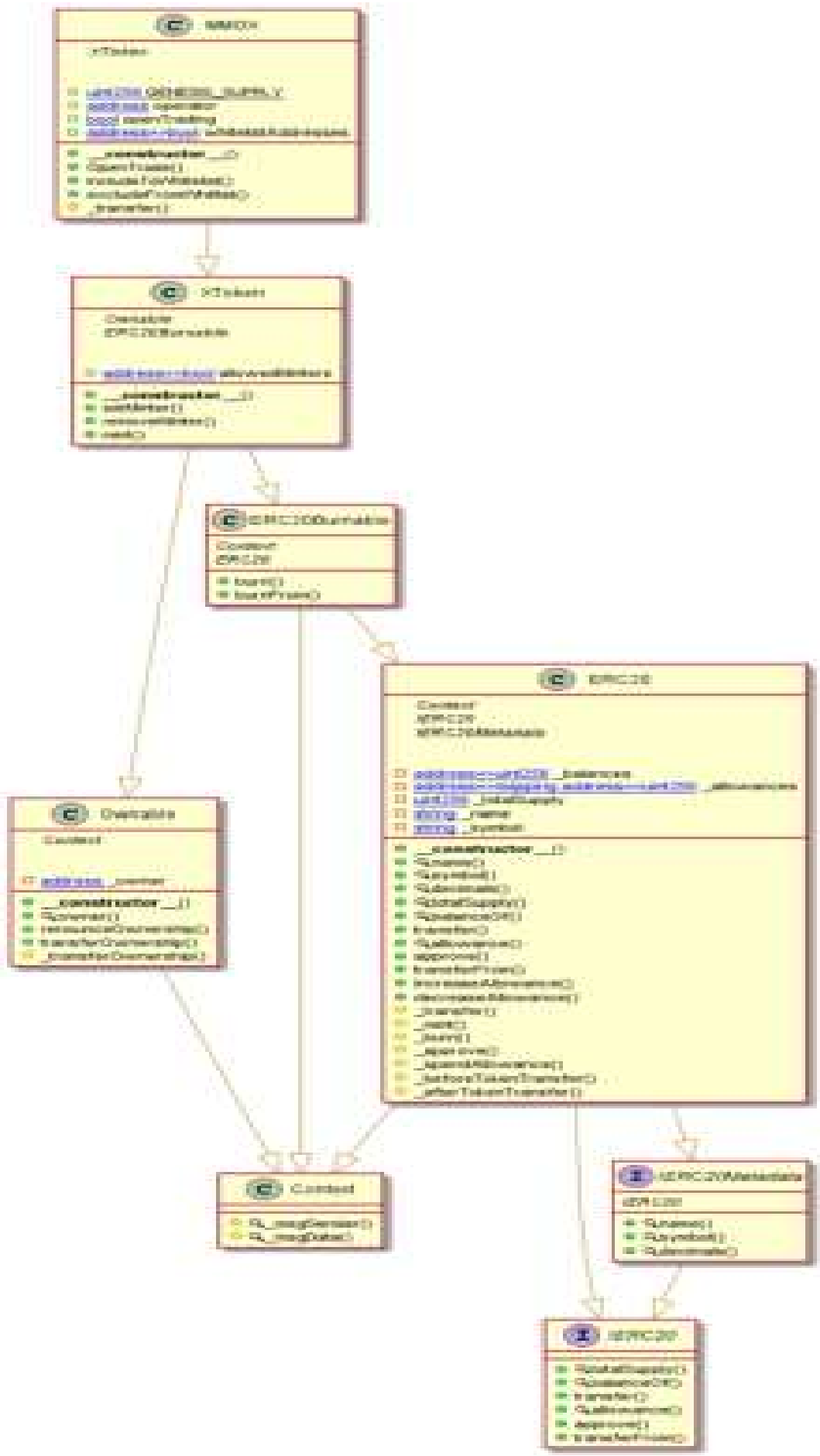
# YToken Diagram

# LOVE Diagram

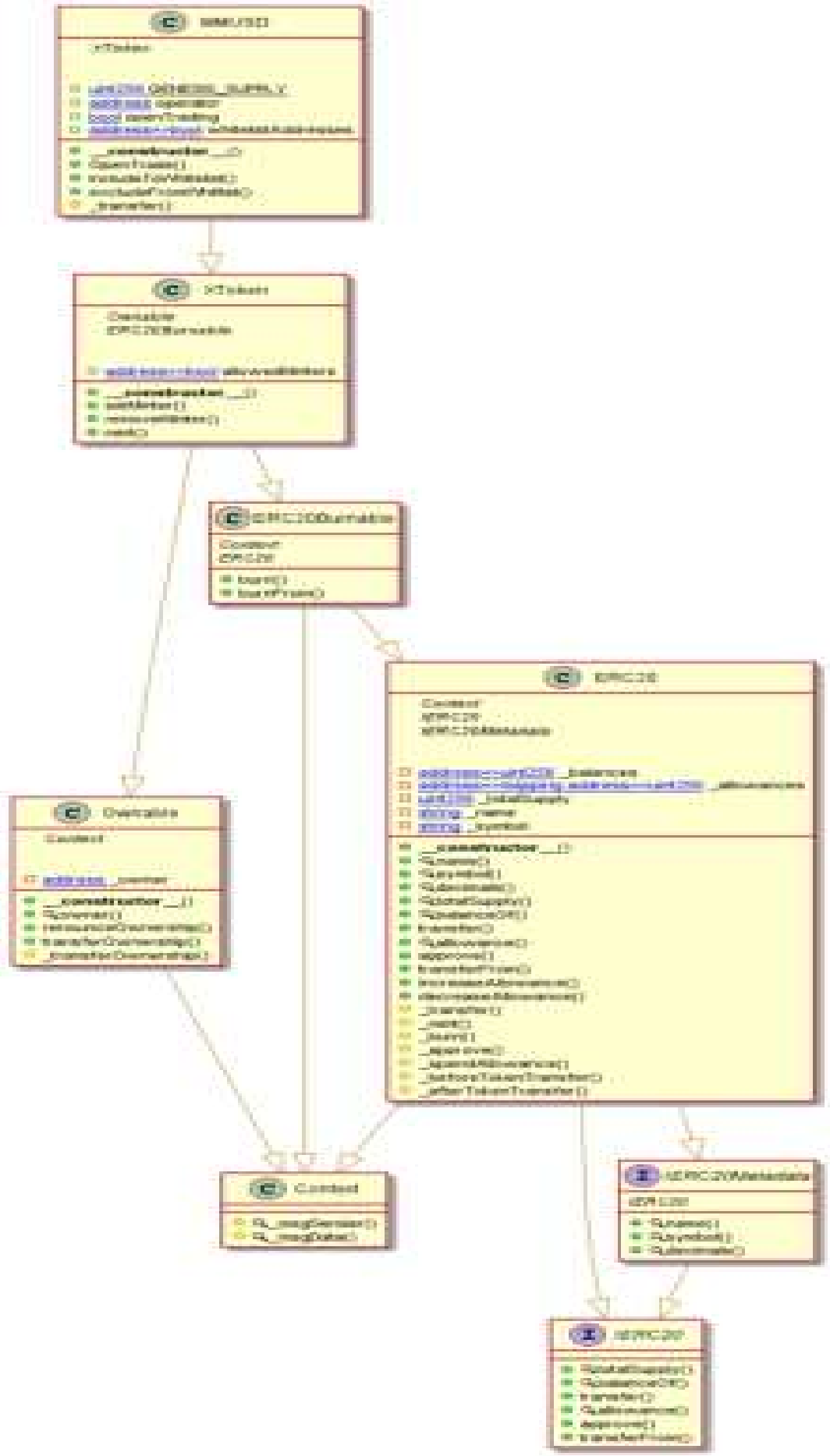
# MMFX Diagram



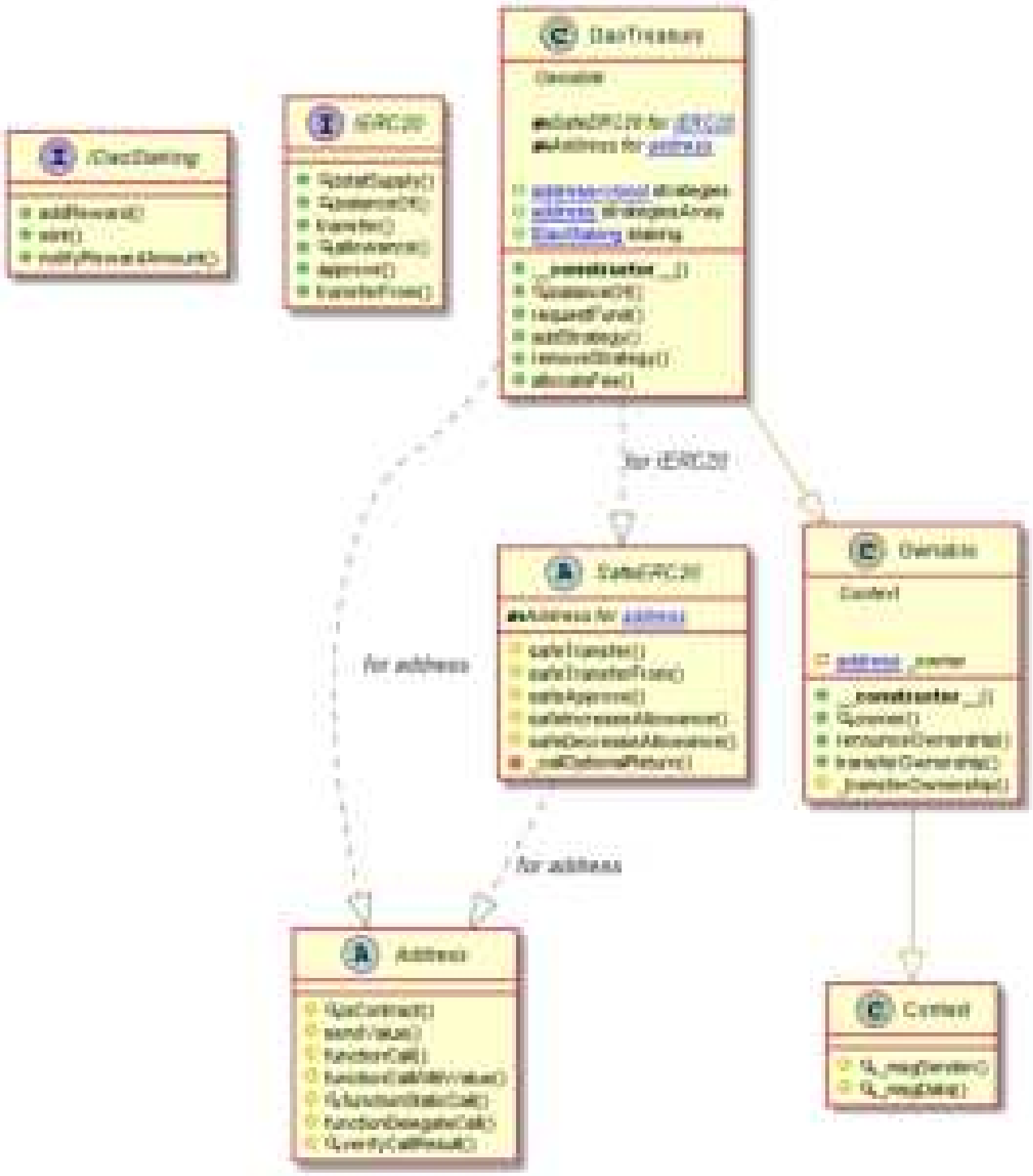
# MMOX Diagram



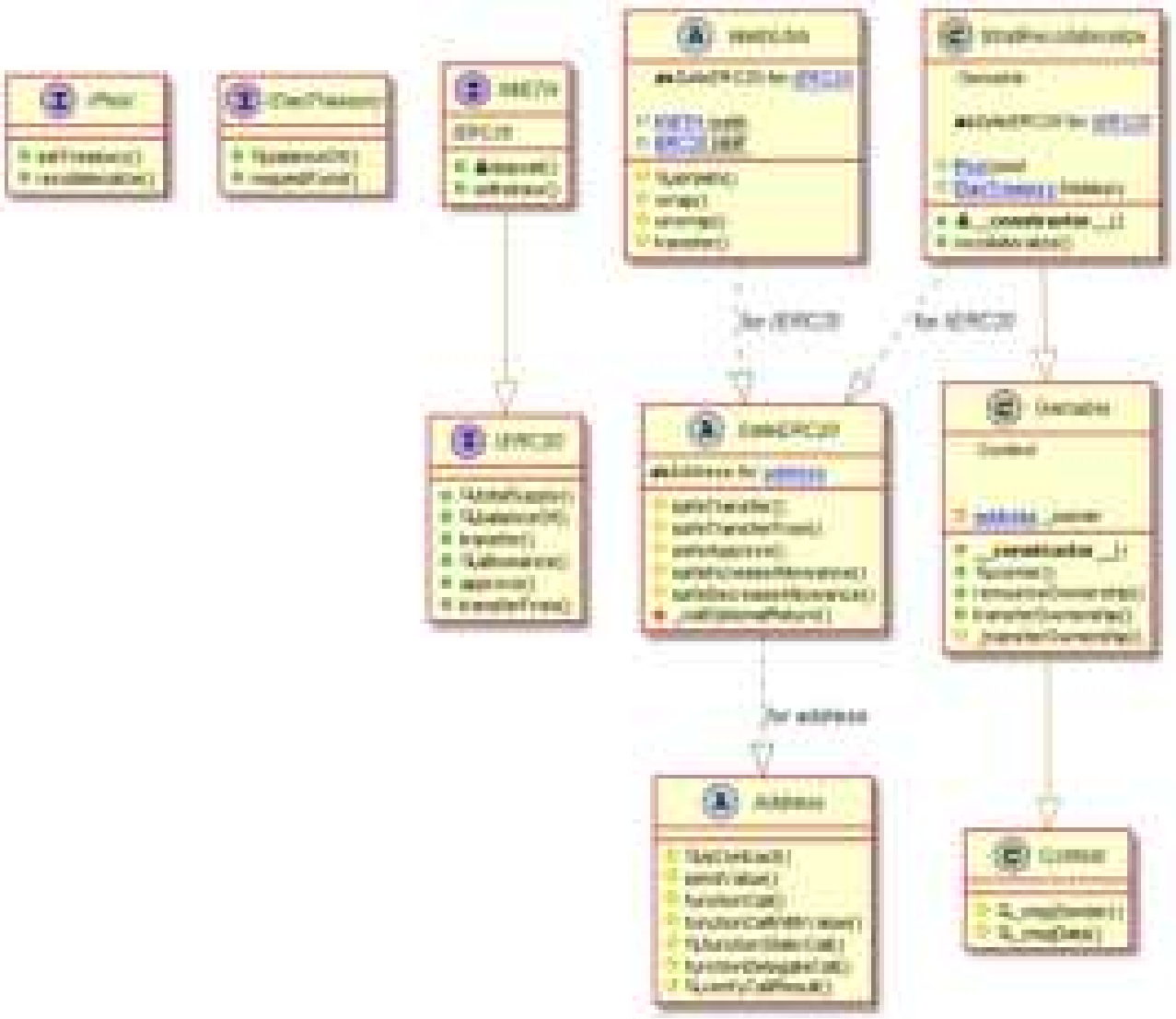
# MMUSD Diagram



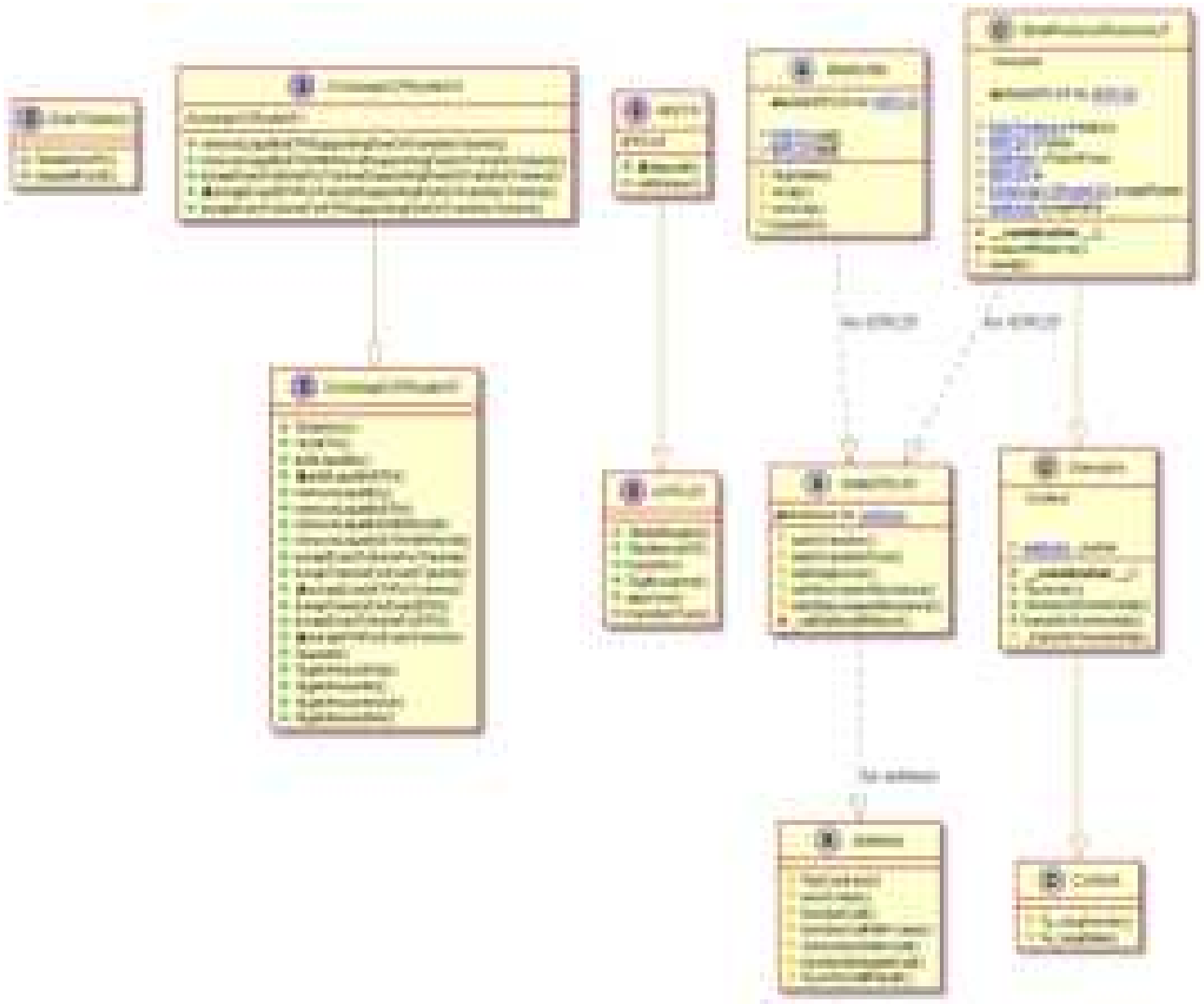
## DaoTreasury Diagram



## StratRecollateralize Diagram

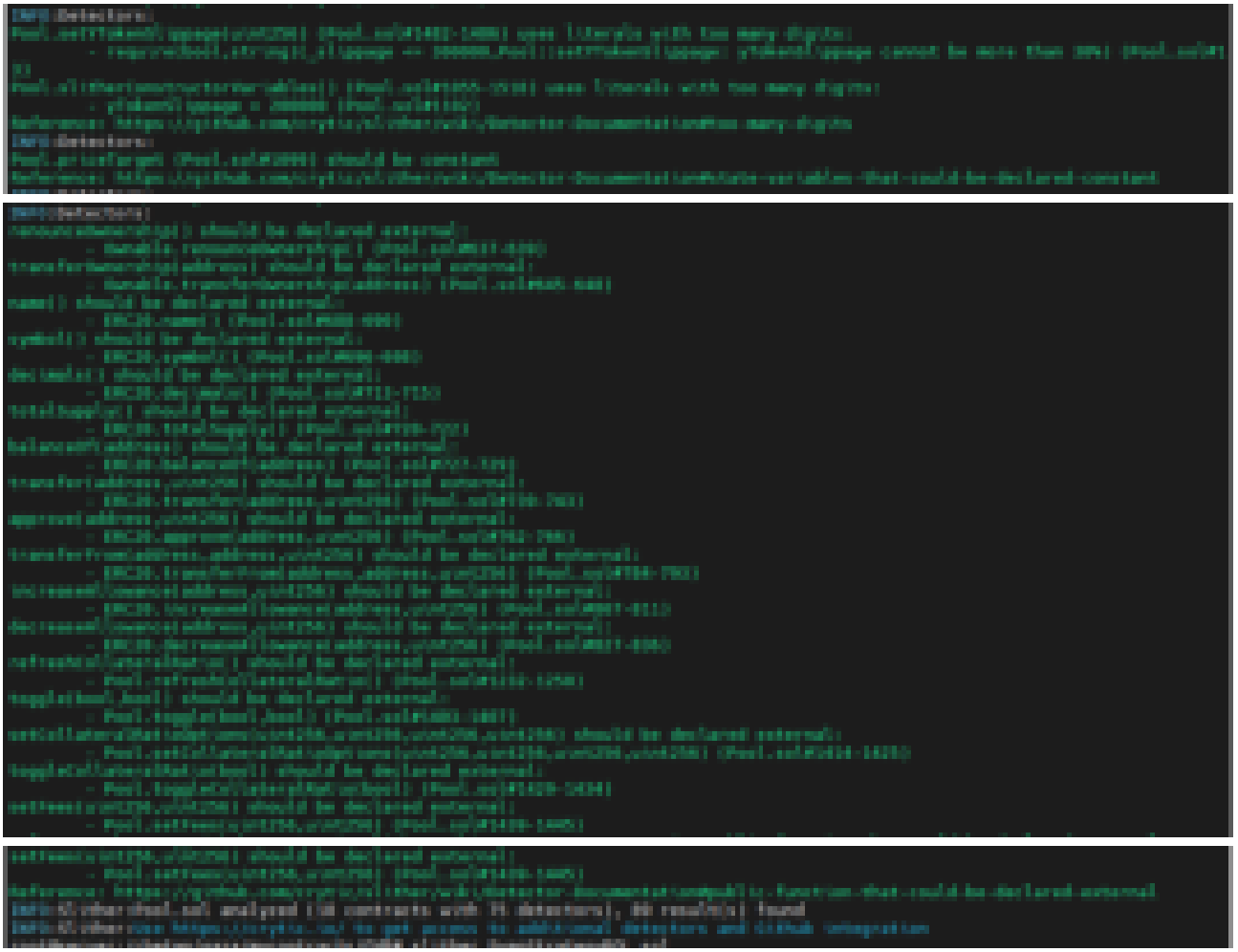


# StratReduceReserveLP Diagram

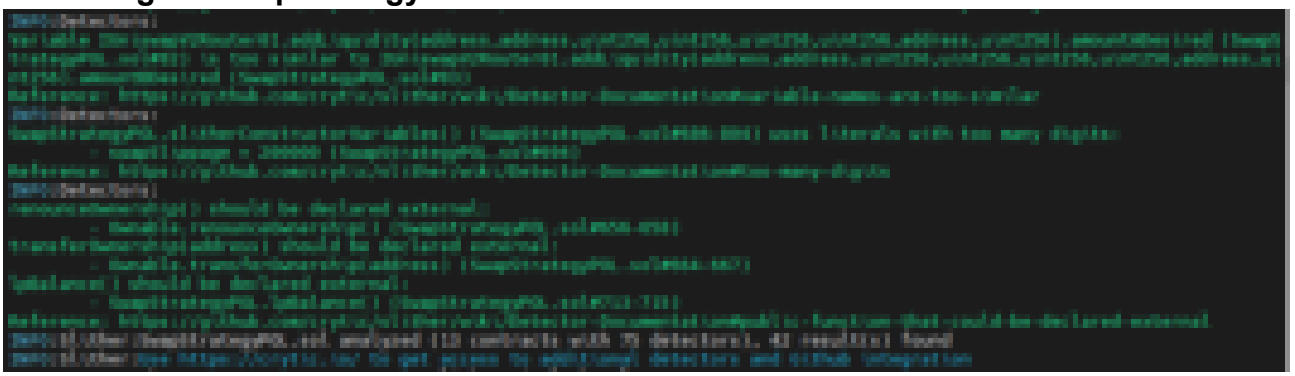


# Slither Results Log

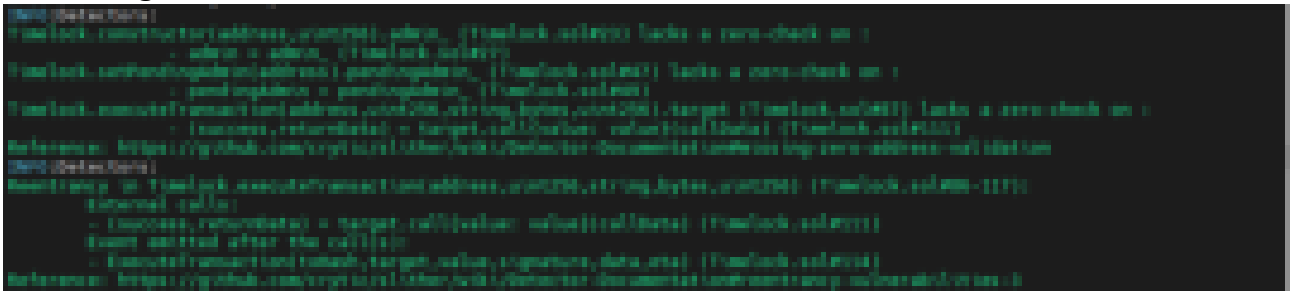
## Slither log >> Pool.sol



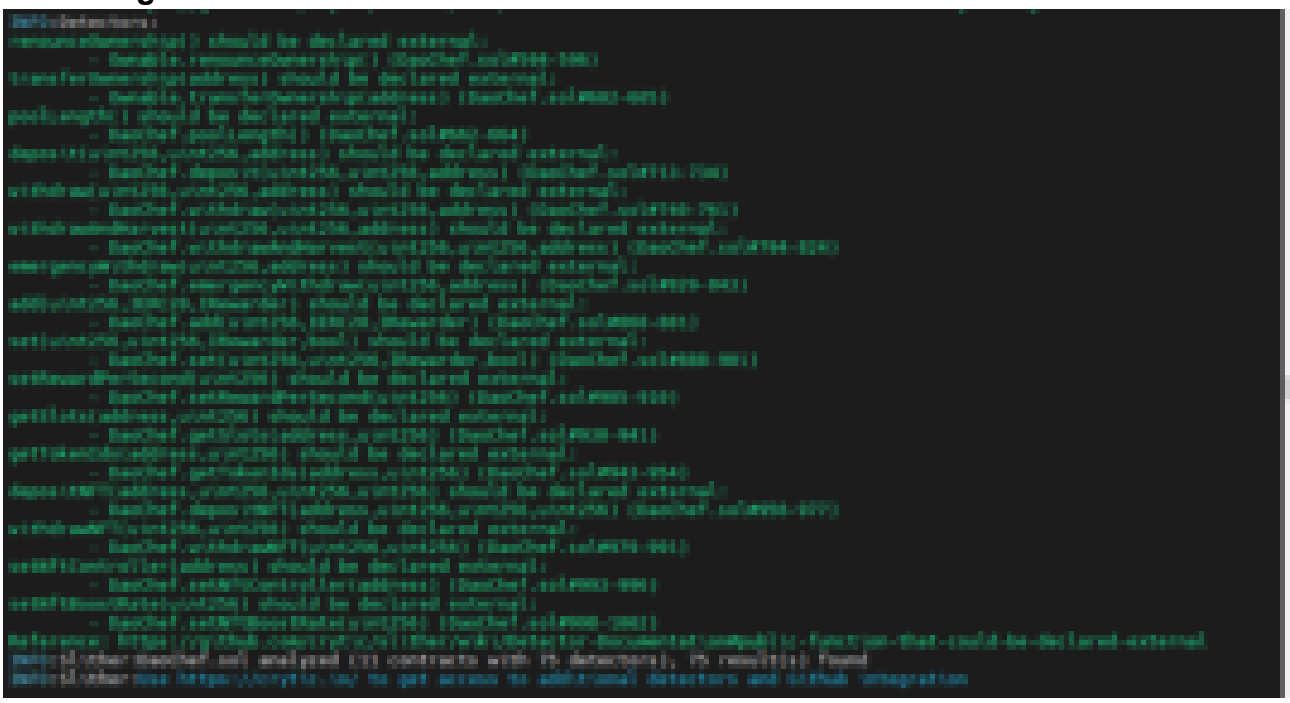
## Slither log >> SwapStrategyPOL.sol



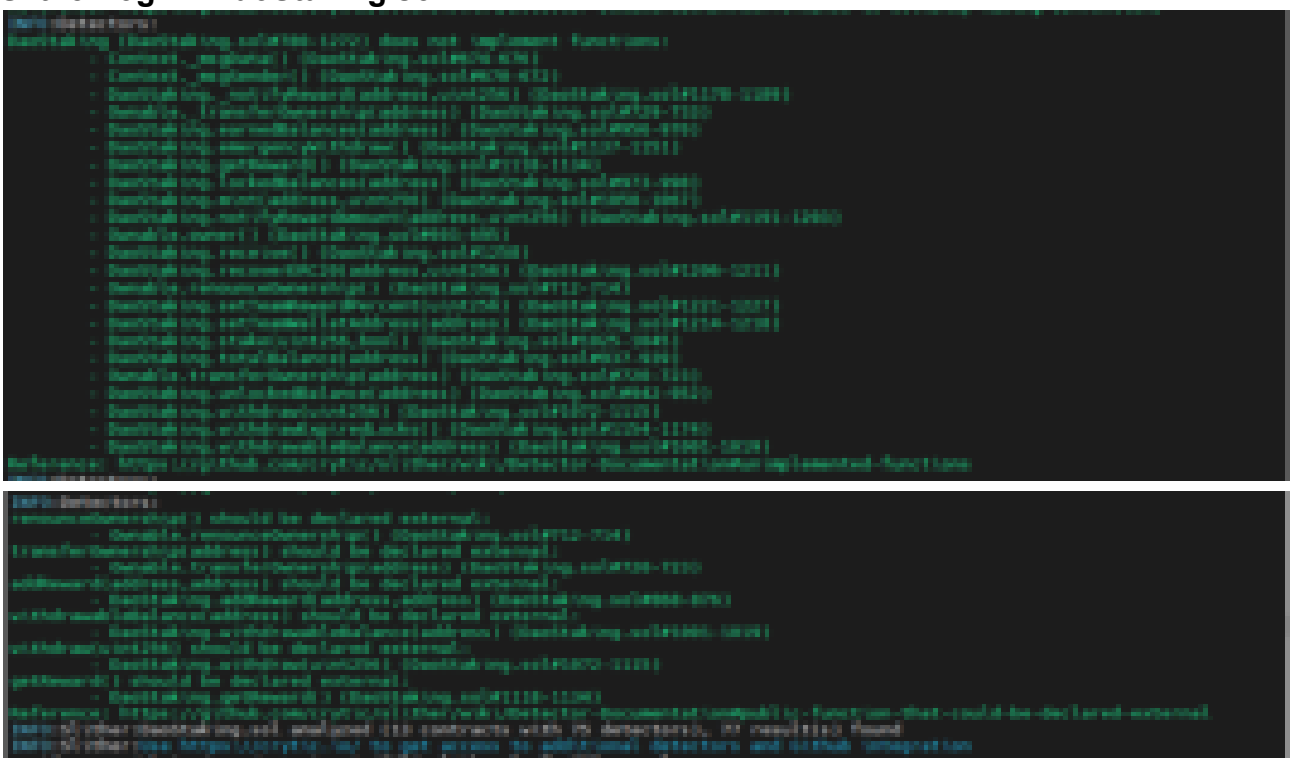
## Slither log >> Timelock.sol



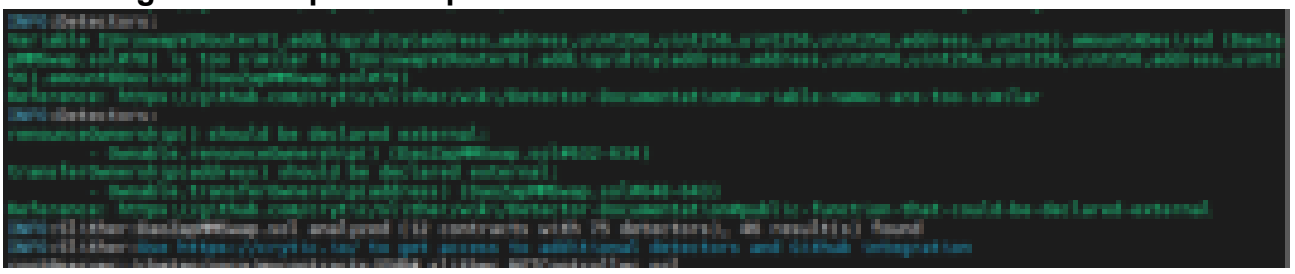
**Slither log >> DaoChef.sol**



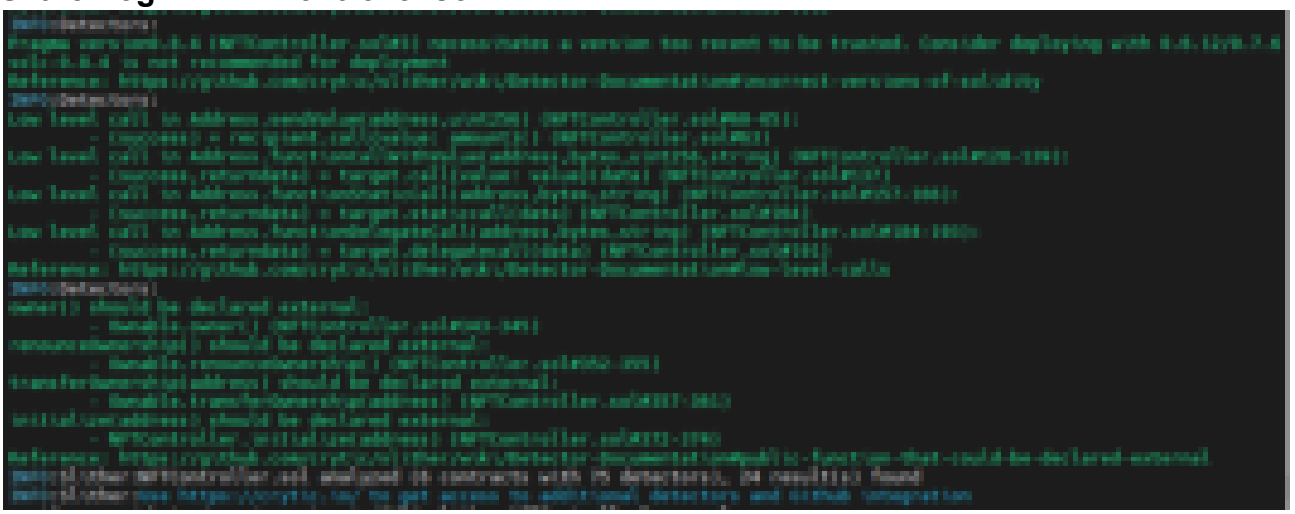
**Slither log >> DaoStaking.sol**



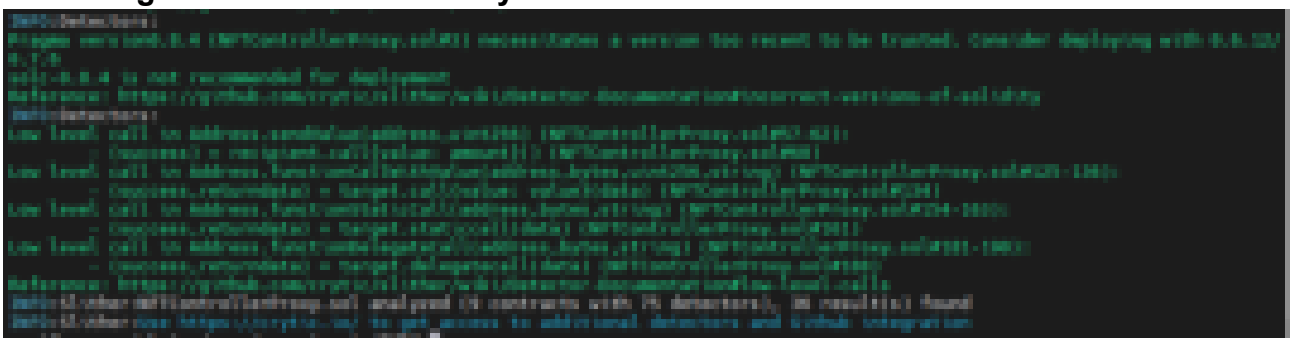
**Slither log >> DaoZapMMSwap.sol**



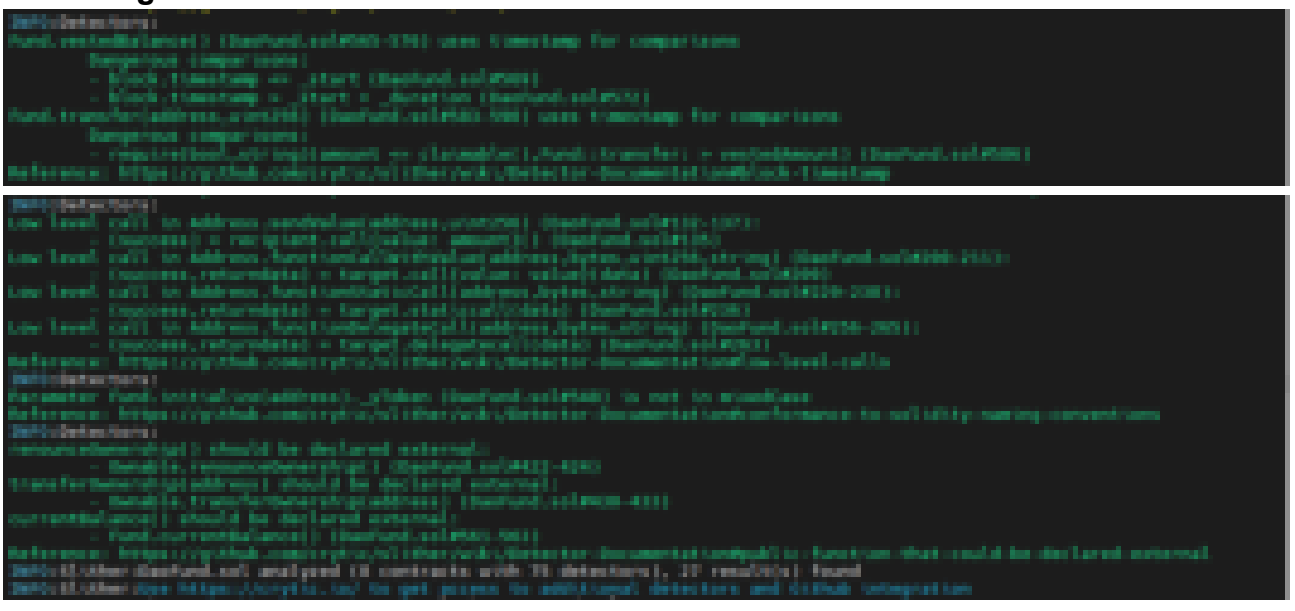
**Slither log >> NFTController.sol**



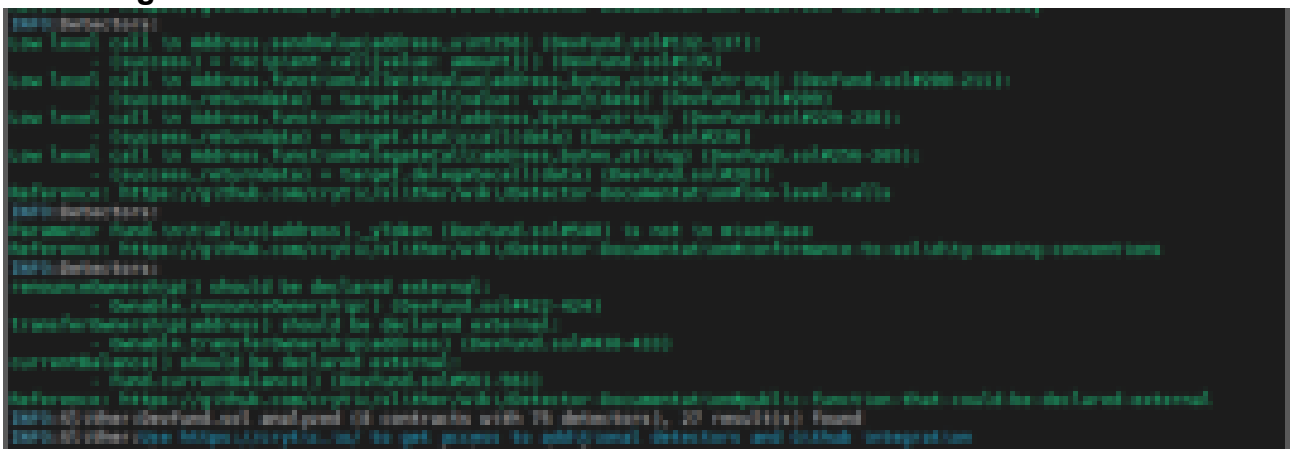
**Slither log >> NFTControllerProxy.sol**



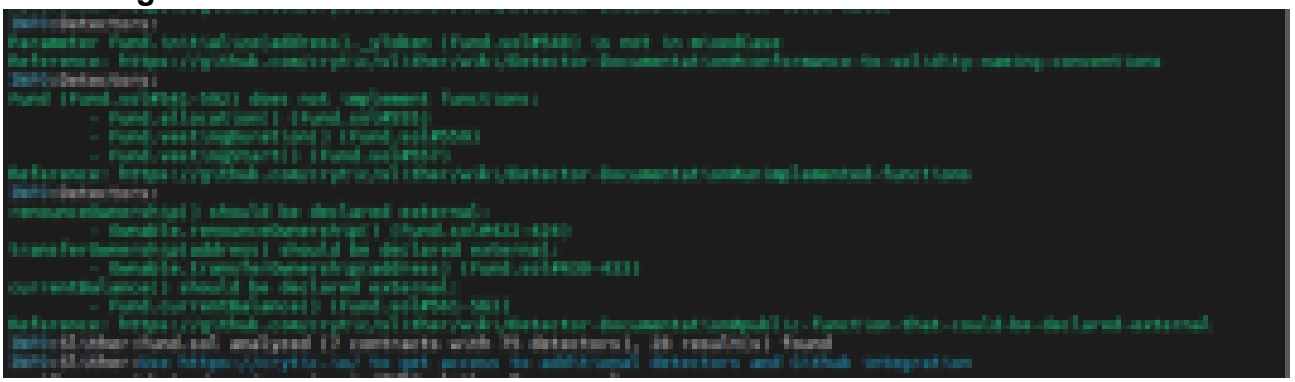
**Slither log >> DaoFund.sol**



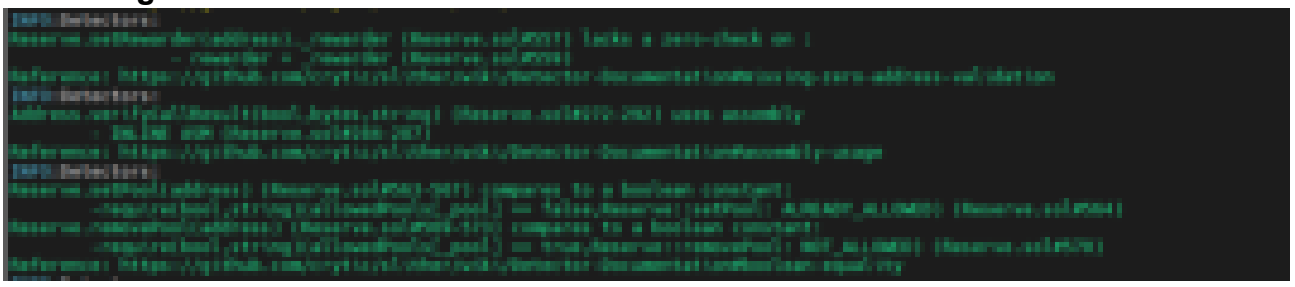
**Slither log >> DevFund.sol**



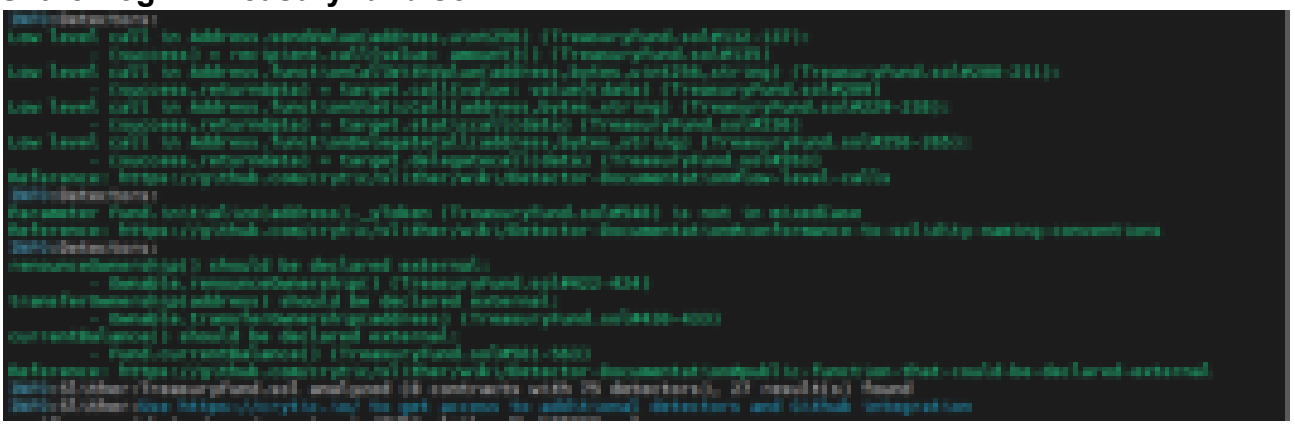
**Slither log >> Fund.sol**



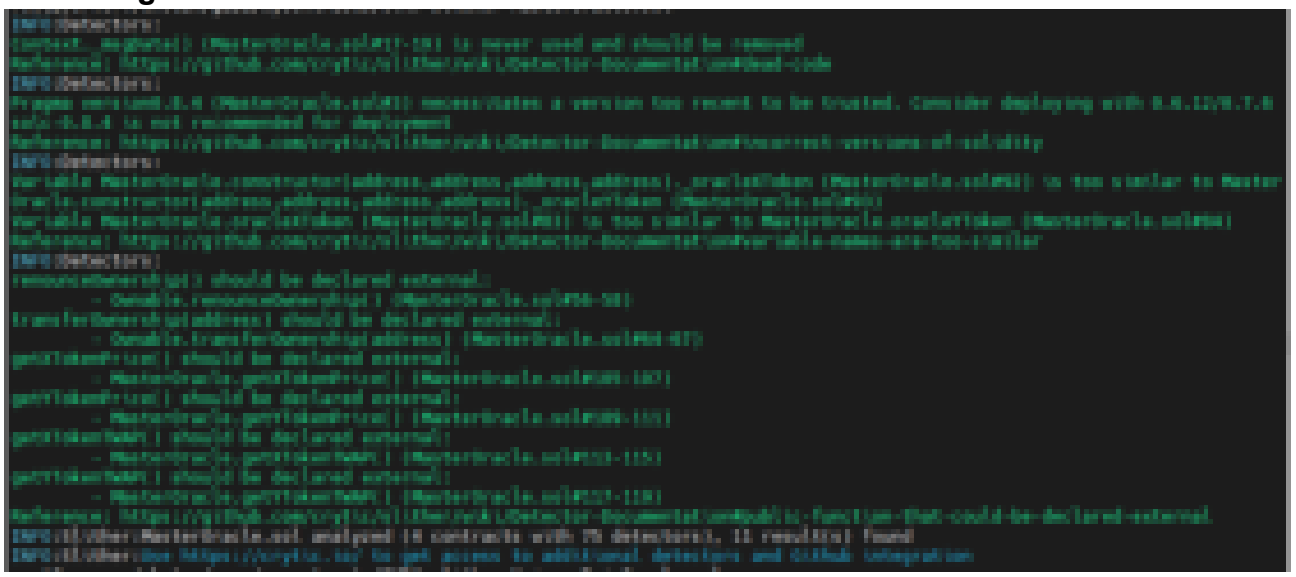
**Slither log >> Reserve.sol**



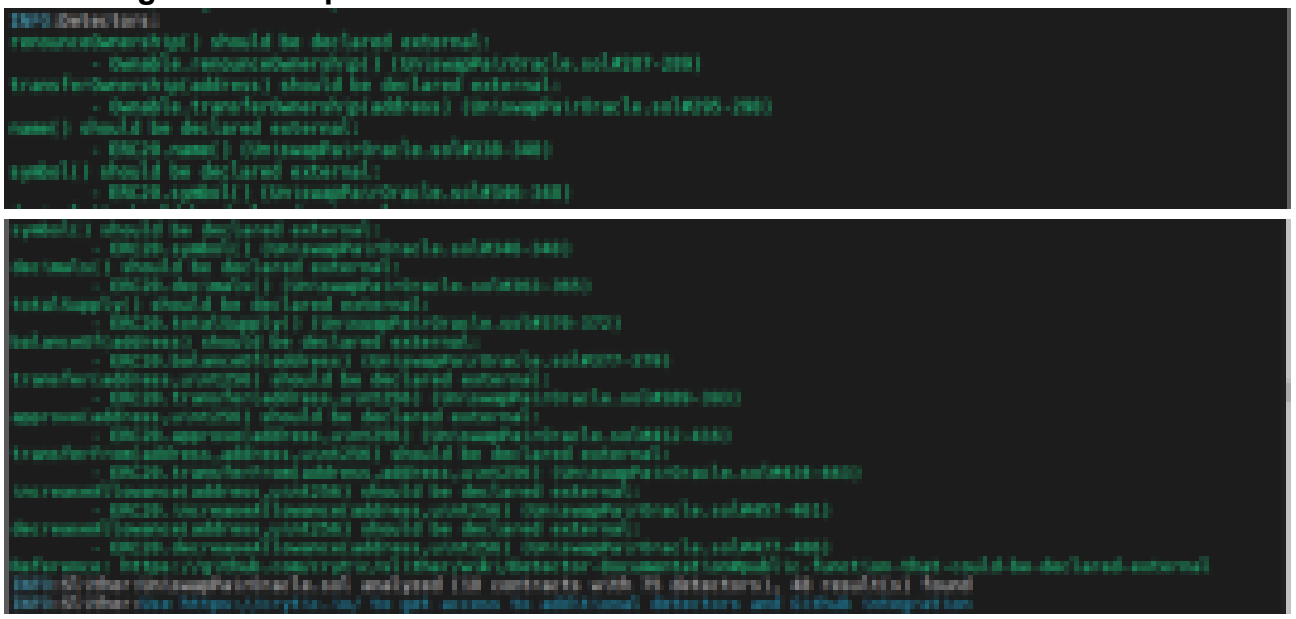
**Slither log >> TreasuryFund.sol**



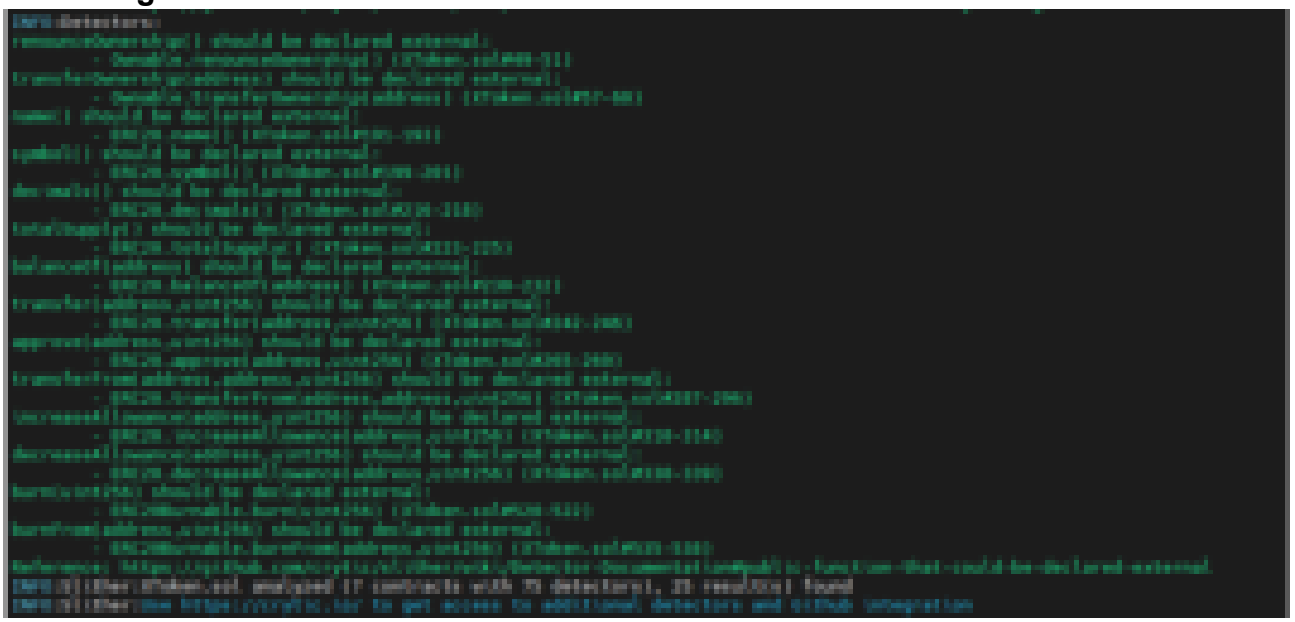
**Slither log >> MasterOracle.sol**



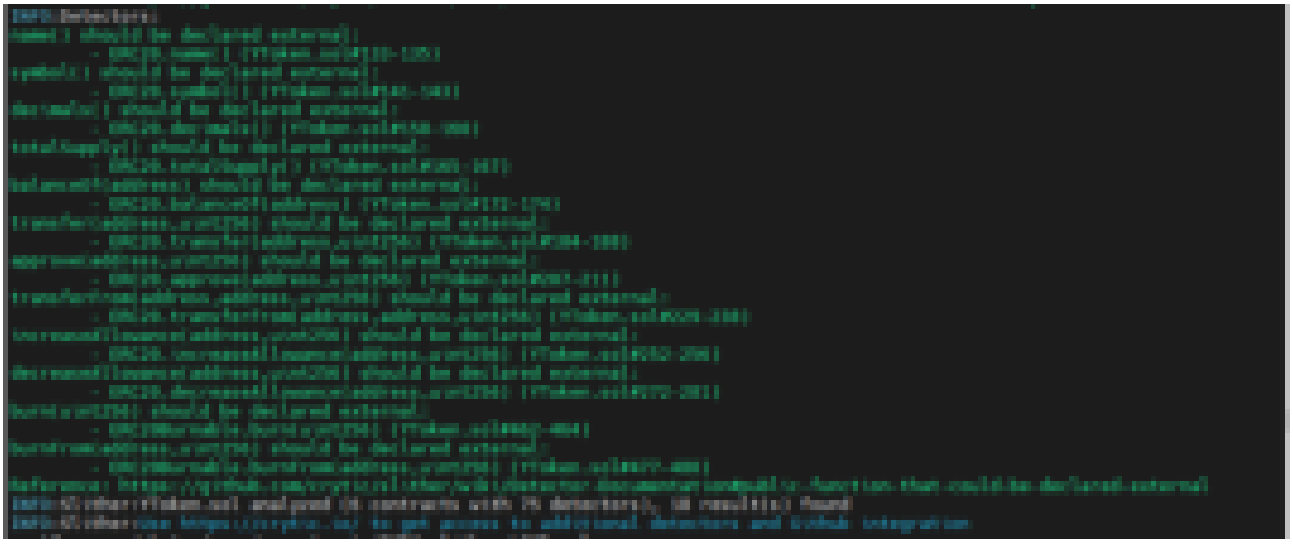
**Slither log >> UniswapPairOracle.sol**



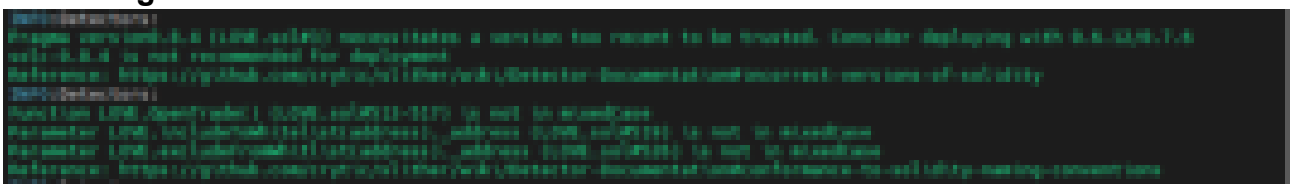
**Slither log >> XToken.sol**

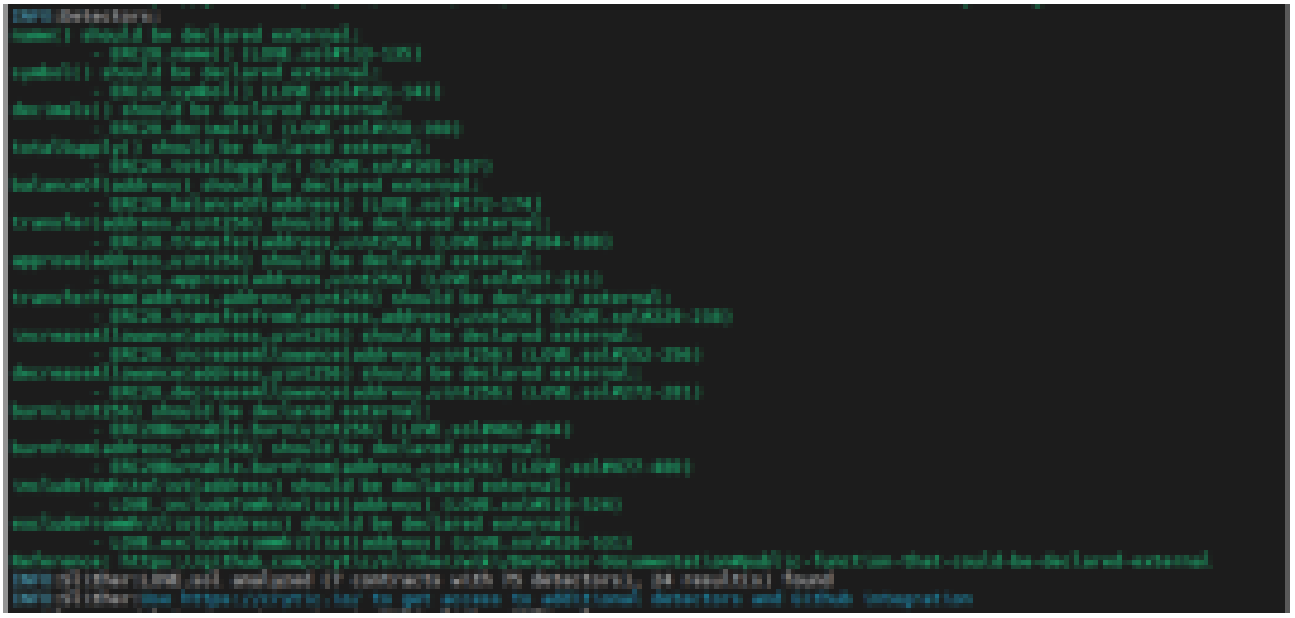


**Slither log >> YToken.sol**

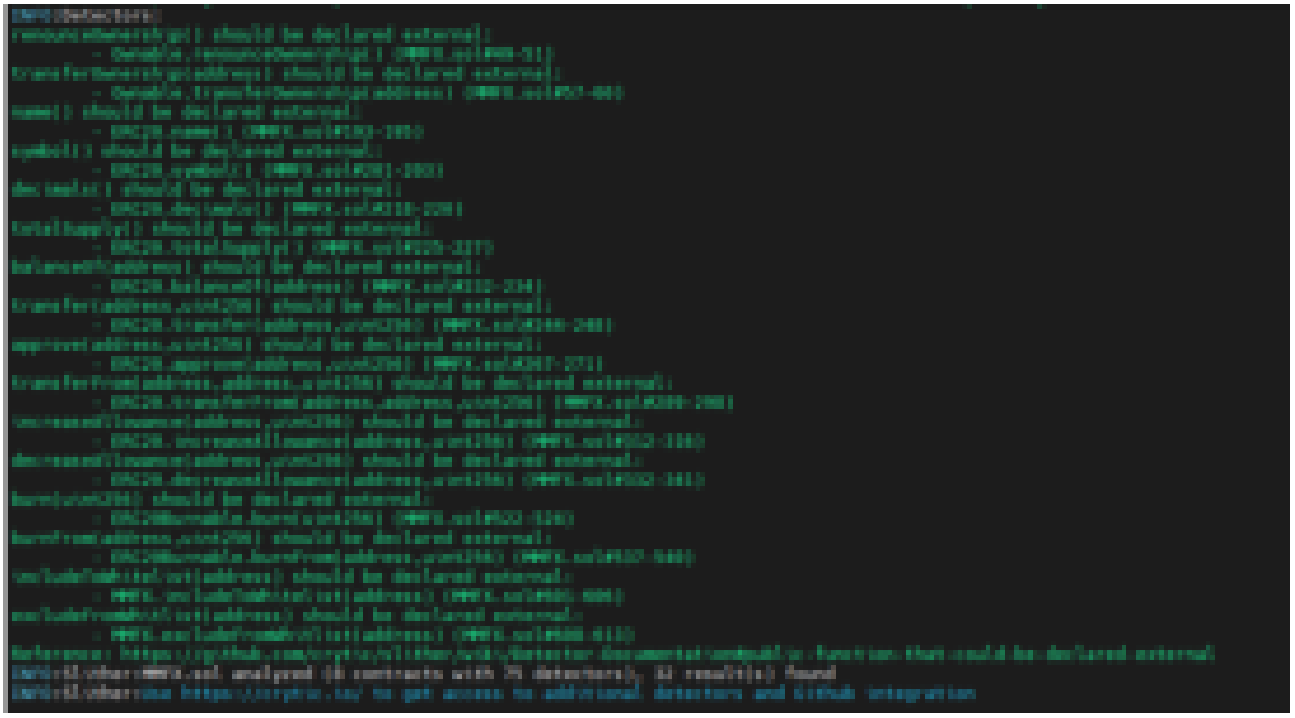


**Slither log >> LOVE.sol**

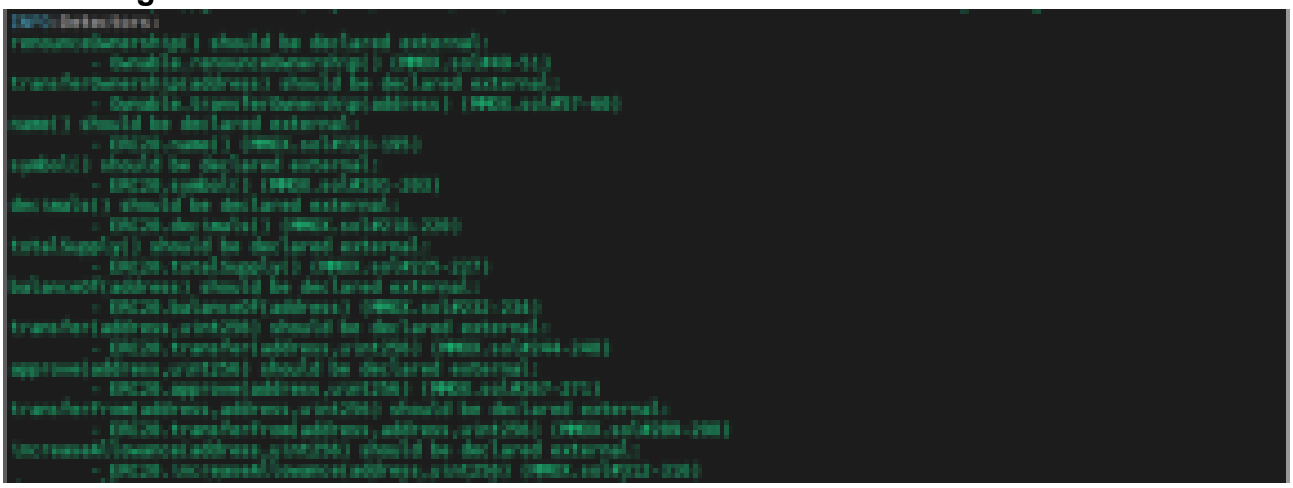


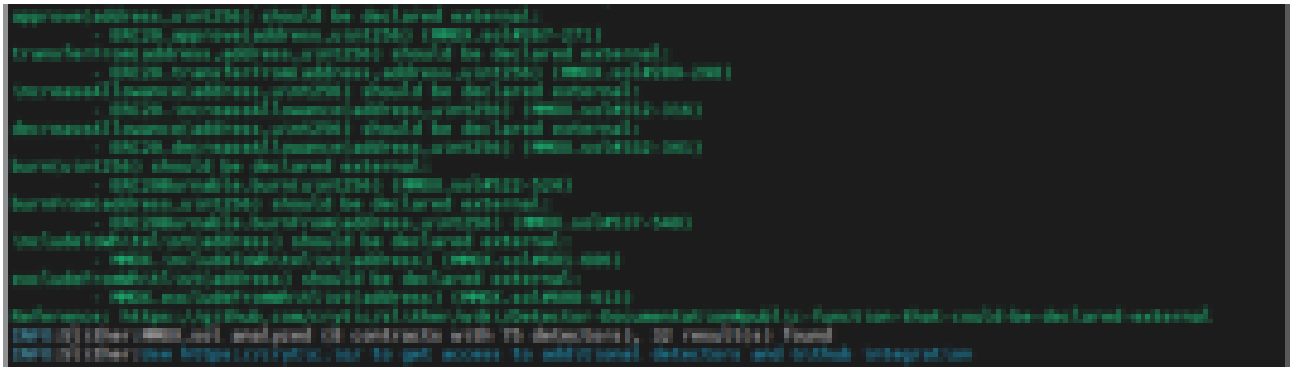


**Slither log >> MMFX.sol**

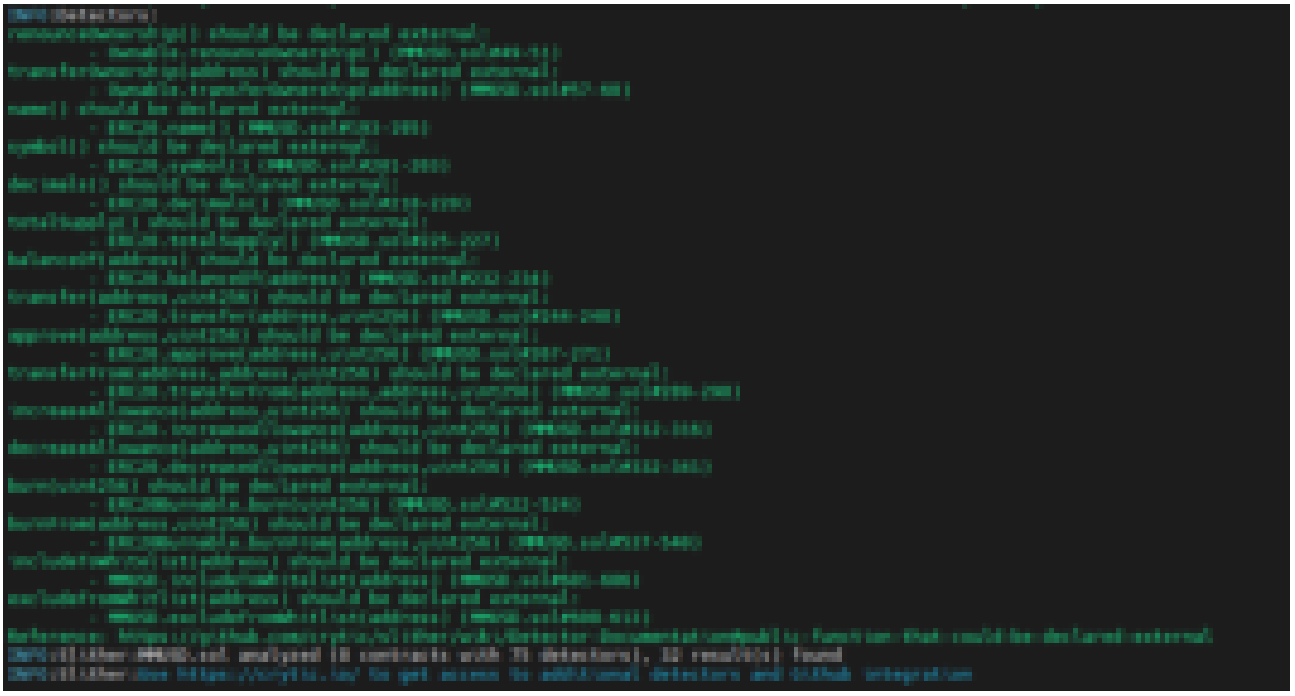


**Slither log >> MMOX.sol**

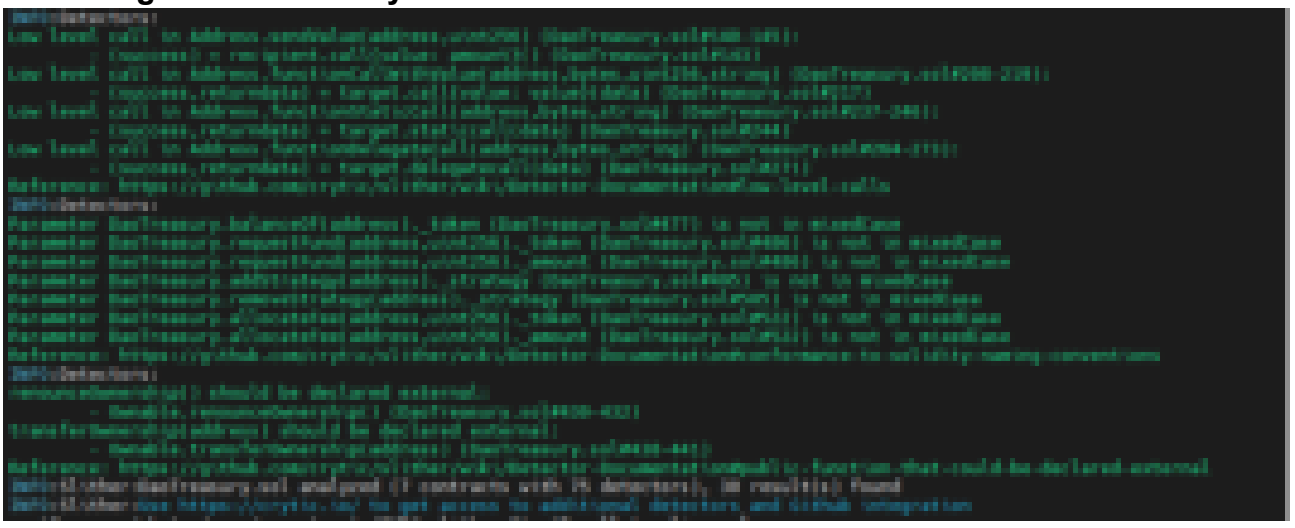




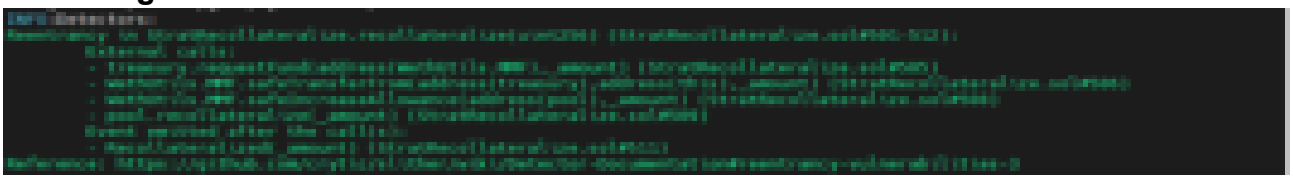
**Slither log >> MMUSD.sol**

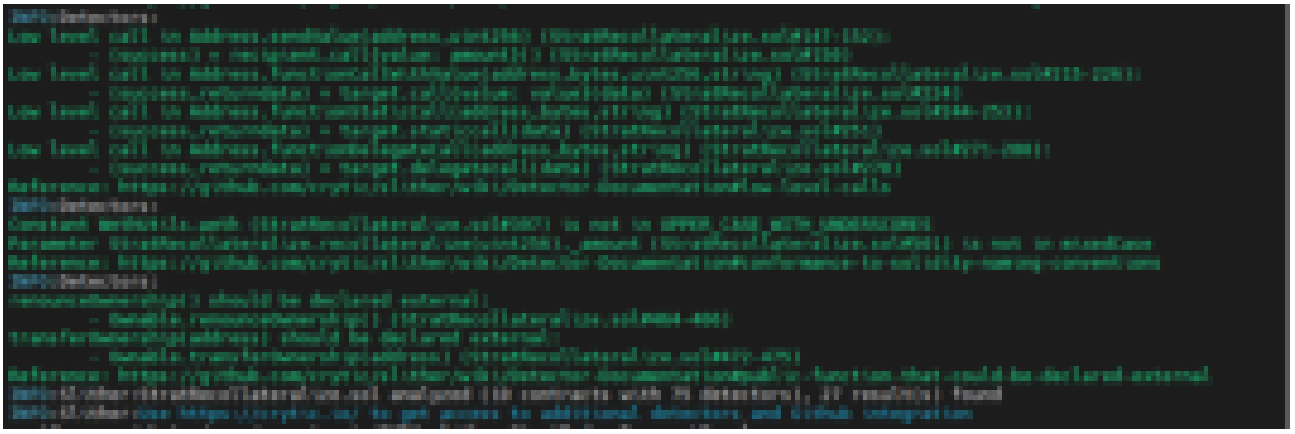


**Slither log >> DaoTreasury.sol**

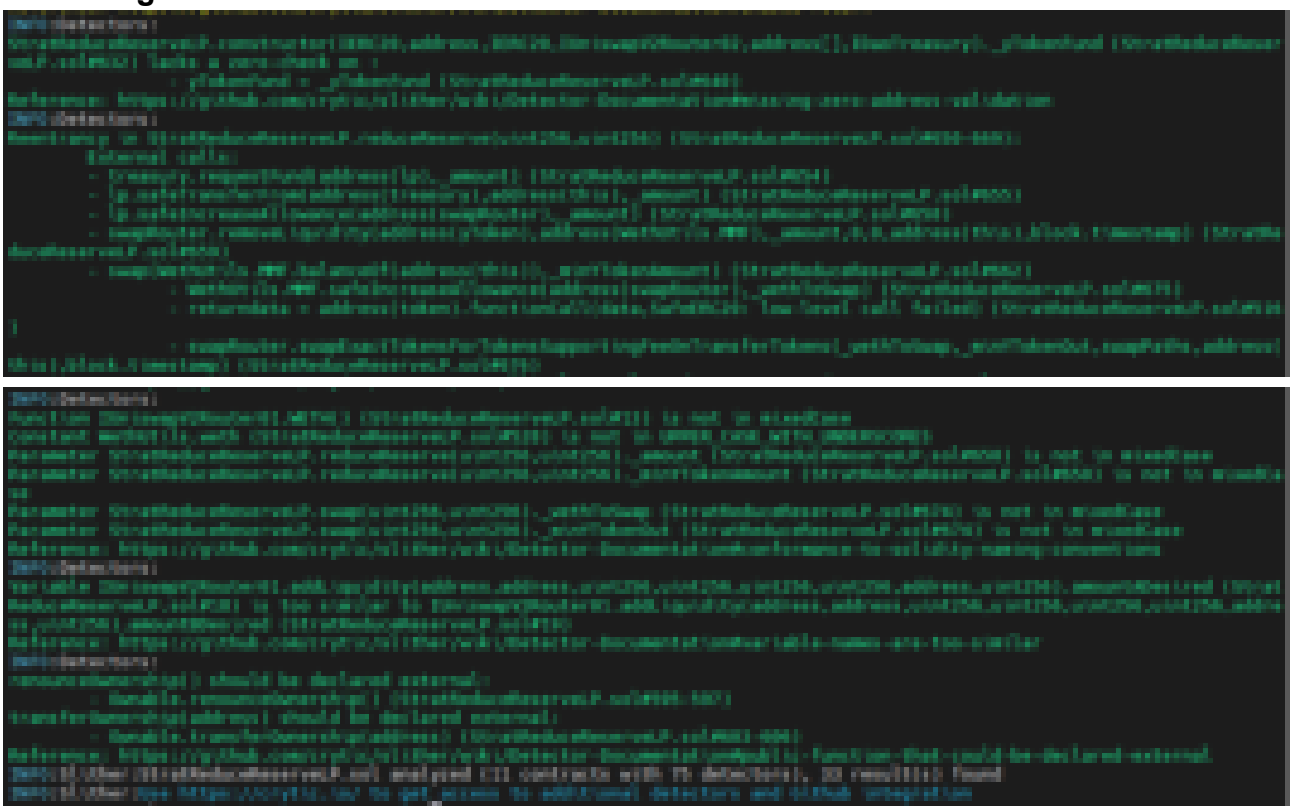


**Slither log >> StratRecollateralize.sol**





**Slither log >> StratReduceReserveLP.sol**






Email: audit@EtherAuthority.io